

(19) 世界知的所有権機関  
国際事務局



(43) 国際公開日  
2001年4月5日 (05.04.2001)

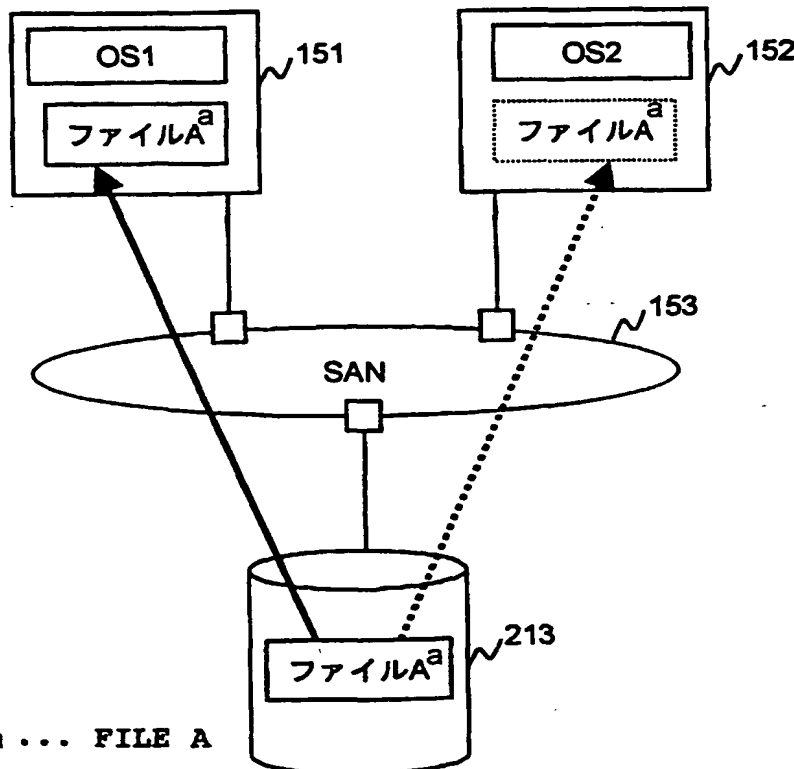
PCT

(10) 国際公開番号  
WO 01/24010 A1

- (51) 国際特許分類: G06F 12/00 [JP/JP]. 小田原宏明 (ODAWARA, Hiroaki) [JP/JP]; 〒185-8601 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社 日立製作所 中央研究所内 Tokyo (JP).
- (21) 国際出願番号: PCT/JP99/05357
- (22) 国際出願日: 1999年9月29日 (29.09.1999) (74) 代理人: 弁理士 作田康夫 (SAKUTA, Yasuo); 〒100-8220 東京都千代田区丸の内一丁目5番1号 株式会社 日立製作所内 Tokyo (JP).
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語 (81) 指定国 (国内): CN, JP, KR, US.
- (71) 出願人 (米国を除く全ての指定国について): 株式会社 日立製作所 (HITACHI, LTD.) [JP/JP]; 〒101-8010 東京都千代田区神田駿河台四丁目6番地 Tokyo (JP). (84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): 高本良史 (TAKAMOTO, Yoshifumi) [JP/JP]. 小原清弘 (OBARA, Kiyohiro) [JP/JP]. 渡辺直企 (WATANABE, Naoki) 添付公開書類:  
— 国際調査報告書
- 2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(54) Title: METHOD OF FILE SHARING AND STORAGE SYSTEM

(54) 発明の名称: ファイル共有方法及びそのストレージシステム



a ... FILE A

(57) Abstract: Multiple hosts using different operating systems (OS) share files stored in disk devices shared among them. In general, if operating systems are different, file systems are also different. One OS cannot see the files managed by the file system of another OS since the storage locations are not understood. According to this invention, the file storage information managed by the file system of a first OS is converted into the file storage information in the form of a second OS so that the second may read the file based on the converted file storage information.

WO 01/24010 A1

[続葉有]



---

(57) 要約:

本発明は、形式の異なるオペレーティングシステム（OS）が稼動する複数のホスト間で共有するディスク装置内に格納されたファイルを各ホスト間で共有する方式に関する。

一般に、OSが異なればそのファイルの管理方式も異なる。従って、その格納場所が分からないため、あるOSの管理方式で管理されているファイルを他のOSから見ることはできない。

そこで、本発明では、第1のOSの管理方式により管理されているファイルの格納場所情報を第2のOS形式による格納場所情報に変換し、上記第2のOSは、その変換した格納場所情報に基づき上記ファイルを読み出す。

## 明 細 書

## ファイル共有方法及びそのストレージシステム

## 技術分野

- 5      本発明は、形式の異なるOSが稼動する複数のホスト間で共有するディスク装置内に格納されたファイルを各ホスト間で共有する方式に関する。

## 背景技術

- 10      計算機は、データを記憶する装置として、コストパフォーマンスが高い磁気ディスク装置を一般に使用する。磁気ディスクは、機械的動作を伴うため処理の時間は10m秒程度とプロセッサの処理速度などと比較すると遅い。プロセッサが高速化されても、ディスクが高速化されないとシステム全体の性能が向  
15      上しないケースも多い。この問題を解決する手段としてディスクアレイがある。ディスクアレイは、W. David Schwaderer, Andrew W. Wilson, Jr., "Understanding I/O Subsystems First Edition" pp. 271-291に述べられているように、複数のドライブにデータを分散して配置するとともに冗長データもドライブに  
20      格納することで、性能と信頼性を向上させる方式である。特に大規模なシステムでは性能と信頼性が同時に要求されるため、ディスクアレイが使用される。

- 近年、ディスクアレイとホストを接続するインタフェースとしてファイバチャネルが有望視されている。SCSI (Small C  
25      omputer System Interface) が数10mまでしか接続距離が伸

ばせないのに対し、ファイバチャネルは数K mまで伸ばすことができる。また、ファイバチャネルには、SCSIに接続可能な装置数の数倍の装置を接続することもできる。ホスト間のデータ通信に用いられているローカルエリアネットワーク（LAN）  
5 N）に対し、近年、ホストと他のデバイス（例えばストレージサブシステム）との間のデータ通信に用いられるストレージエリアネットワーク（SAN）が注目されている。ファイバチャネルは、ストレージエリアネットワークの通信媒体としても使用される。ファイバチャネルは、規格化されており、規格に合ったデバイスやホストであれば、ストレージエリアネットワークに接続することができる。例えば、ファイバチャネル  
10 インタフェースを持つディスクアレイと複数のホストを接続することが可能となる。

複数のホスト間でデータを共有する方法を開示するものとして、特開平9-325905号公報（以下「文献1」という。）  
15 がある。文献1は、メインフレームで採用されているデータフォーマットであるCKD（Count Key Data）フォーマットとUNIX等で採用されているデータフォーマットであるFBAフォーマットに変換する機能を有するディスク装置を開示する。  
20 ここでフォーマットとは、ホストとディスク装置との間で転送されるデータの最小単位の構造を意味する。CKDは可変長形式であり、FBAは固定長形式である。CKD形式で格納されたディスクには、ユーザデータとデータ長と検索に使用するキーが格納される。例えば、2048バイトのユーザデータをディスクに書き込む場合には、データ長とキー値と2048バイ  
25

トのデータが一度に書き込まれる。一方、F B A形式では、同じように2048バイトのデータをディスクに書き込む場合には、2048Kバイトのデータを512バイトの固定長に区切りられたブロック毎にディスクに格納される。F B Aでは、データ長やキーは格納されない。このように同じディスク装置でも、ホストの種類によってディスク装置内のデータ格納形式が異なるため、異なるホスト間でデータを共有することができない。文献1は、この問題を解決するために、C K D形式で格納されたディスクに対して、F B A形式でアクセスした場合、C K Dのデータ部をF B A形式のブロックに変換する機能を有している。

他のファイル共有方式としては、特開平9-179758号公報（以下「文献2」という。）がある。文献2には次のような技術が開示される。書き込み自在な記録媒体中のデータ共有領域にデータを書き込む。それと共に、その記録媒体に、そのデータをアクセスする際に必要なデータ管理情報を複数種類のオペレーティングシステムのそれぞれから参照可能な複数のファイルテーブルを書き込む。

## 20 発明の開示

上述したS A Nの目的の一つは、S A Nに接続される様々のO S（オペレーティングシステム）のプラットフォームから、同じくS A Nに接続されるストレージシステムに自由にアクセス出来るようにすることにある。従って、異なるO Sのプラットフォームでファイルを共有できるようにすることが望まれる。

ファイバチャネルで接続された装置は、規定されている通信  
プロトコルに従えば、そのファイバチャネルに接続された他の  
装置に対して、物理的にアクセスすることができる。例えば、  
5 ホストとディスク装置とをファイバチャネルで接続した場合、  
ホストは、ディスク装置の物理的な領域、具体的にはディスク  
の最小アクセス単位であるセクタと呼ばれる固定長ブロック領  
域に読み書きすることができる。

しかし、ディスク領域にどのようなデータをどのように格納  
するかは、通信プロトコルではなく、ホスト上で稼動している  
10 オペレーティングシステムが決定する。従って、単に異なるO  
Sを搭載するホストとストレージシステムとをファイバチャネ  
ル等で物理的に接続しても、ストレージシステム内のファイル  
を共有することはできない。互いに、互いのOSで管理される  
ファイル情報が分からないからである。

15 詳細には、オペレーティングシステムの中のファイルシス  
テムと呼ばれるディスク領域の管理プログラムが、ディスク内の  
領域を管理する。例えば、ホスト上のアプリケーションプログ  
ラムがファイルAを作成したいと要求した場合、ファイルAの  
データは、ファイルシステムが決定したディスク内のいずれか  
20 の領域に格納される。ファイルAのデータがディスク内のどこ  
に格納されたかをアプリケーションプログラムは知る必要はな  
い。例えば、ディスク装置に、第1のオペレーティングシス  
テムが搭載された第1のホストと、第1のオペレーティングシ  
ステムとは異なる第2のオペレーティングシステムが搭載され  
25 第2のホストをファイバチャネルで接続されているとする。こ

の場合、第1のホストからも第2のホストからも、そのディスク装置に物理的にアクセスすることはできるかもしれない。しかし、一般に、ファイルシステムのディスク領域管理方法はオペレーティングシステムによって異なるため、第1のオペレーティングシステムのファイルシステムが格納したファイルを、第2のオペレーティングシステムから読むことはできない。どのファイルが何処に、どのように格納されているか分からないからである。例えば、第1のオペレーティングシステムとしてUNIXを、第2のオペレーティングシステムとして、マイクロソフト社のMicrosoft Windowsを想定した場合、UNIXのファイルシステムが格納したファイルをMicrosoft Windowsから読むことはできない。

文献1に記載されているディスクアレイ装置は、データフォーマットの変換機能しか有していない。従って、データフォーマットの変換を行いたいデータが特定されている場合には、文献1の方法によりメインフレームとオープン系ホスト間でデータの共有が可能である。しかし、ユーザは一般にファイルと呼ばれるデータの集合に対してアクセスしており、また個々のデータはディスク装置内に連続して格納されているわけではない。従って、他のホストが作成したファイルにアクセスしようとする場合に、そのファイルがディスク装置上のどこに格納されているかを知らなければ、文献1の方法では、そのファイルを読むことはできない。。

一方、文献2が開示する方法では、各ホストが自ホスト内のファイルシステムと同じ形式で格納された管理情報の位置をあ

あらかじめ知っておく必要がある。つまり、記憶媒体内に複数のファイルシステムの管理情報が格納されていることを知っているホストしかアクセスすることができない。これは、一般的なファイルシステムが、メディア内の特定の決められた位置に管理情報を配置していることが前提となっているためである。例えば、記録媒体の第2ブロックの先頭から管理情報が配置されているといったように、ファイルシステムが独自に決めた特定領域がある。もし、この領域にまったく異なるデータが書き込まれた場合には、ファイルシステムは正常な動作ができなくなる。また、自ホストに合ったファイルシステムの管理情報が格納された位置をホスト側で管理しなくてはならないので、例えば記録媒体内の管理情報の格納場所等を変えた場合には、それにあわせてホスト側のデータの変更も行わなければならないという管理上の問題もある。逆に、記録媒体に接続されるコンピュータに搭載されるOSを変更する場合には、記録媒体内の管理情報を書き替えなければならい。OSにより管理情報を格納するのに必要な料率な大きさ等は異なるので、その変更をする際に、領域の確保等困難を伴うこともあり得る。さらに、複数のファイルシステムの管理情報を単一のメディアに格納しなくてはならので、使用可能な領域が少なくなるという問題もある。

そこで、本発明の目的は、アクセスしたいデータがディスク装置上のどこに格納されているかをホストが知らない場合であっても、また、ファイルシステムの管理情報が格納された位置をホスト側で管理しなくても、複数の異なるプラットフォームのホスト間でファイル共有を行うことができるディスク装置を



提供することにある。

上記目的を達成するために、本発明では、ディスクアレイ制御装置内に、ファイルシステムの形式を変換するプログラムと、ホストと稼動しているファイルシステムとの対応表とを持つ。

- 5    ホストからファイルシステムの管理情報へのアクセスが発生した場合、前記対応表から、ファイルシステムに応じたファイル管理情報にアクセスする。

#### 図面の簡単な説明

- 10    第1図は、本発明の実施例1のストレージシステムを用いたシステム全体の構成を説明する図である。

第2図は、ホスト1及びディスクアレイ制御装置における、データ記憶の階層構造を説明するための図である。

- 15    第3図は、異なるOSのマルチプラットフォームによるファイル共有の概念を説明するための図である。

第4図は、第1のOSのファイルシステムのディスク領域管理方法を説明するための図である。

第5図は、第2のOSのファイルシステムのディスク領域管理方法を説明するための図である。

- 20    第6図は、第1のOSの管理構造を説明するための図である。

第7図は、第2のOSの管理構造を説明するための図である。

第8図は、第1のOSのファイルシステムと第2のOSのファイルシステムとの間でファイルを共有する方式を説明するための図である。

- 25    第9図は、ホストテーブルの構成を示す図である。

第 10 図は、変換プログラムの構成を示す。

第 11 図は、パーティション変換プログラムのフローを示す。

第 12 図は、ファイルシステム変換プログラムのフローを示す。

5 第 13 図は、パーティション変換プログラムの変換結果を説明するための図である。

第 14 図は、ファイルシステム変換プログラムの変換結果を説明するための図である。

第 15 図は、変換プログラムを実行後のディスクシステムの  
10 動作を説明するための図である。

第 16 図は、ディスクアレイ読み込みプログラムのフローを示す。

第 17 図は、本発明の実施例 2 のディスク上のデータ格納形式を説明するための図である。

15 第 18 図は、本発明の実施例 2 のディスクアレイ読み込みプログラムのフローを示す。

第 19 図は、本発明の実施例 3 のデータ共有方式を説明するための図である。

第 20 図は、実施例 3 のディスクアレイ読み込みプログラムの  
20 フローチャートを示す。

第 21 図は、本発明の実施例 4 のストレージシステムを用いたシステム全体の構成を説明する図である。

第 22 図は、本発明の実施例 3 のファイルシステム変換プログラムのフローチャートを示す。

25 第 23 図は、変換制御プログラムのフローチャートを示す。

第 2 4 図は、本発明の実施例 5 のストレージシステムを用いたシステム全体の構成を説明する図である。

第 2 5 図は、ロックテーブル 2 4 0 1 の構成を示す。

第 2 6 図は、ホスト内で稼動するアプリケーションが、ディスクアレイ制御装置に対してロックを掛ける／はずす制御を説明するための図である。

第 2 7 図は、ロック制御プログラムのフローチャートを示す。

第 2 8 図は、ロック処理のフローチャートを示す。

第 2 9 図は、アンロック処理のフローチャートを示す。

10 第 3 0 図は、本発明の実施例 3 のディスクアレイ書き込みプログラム 2 4 0 3 のフローチャートを示す。

第 3 1 図は、ロック検査のフローチャートを示す。

第 3 2 図は、本発明の実施例 6 のストレージシステムを用いたシステム全体の構成を説明する図である。

15 第 3 3 図は、本発明の実施例 6 のディスクアレイ読み込みプログラムのフローチャートを示す。

第 3 4 図は、同期制御のフローチャートを示す。

第 3 5 図は、割込制御プログラムのフローチャートを示す。

20 第 3 6 図は、ディスク更新制御プログラムのフローチャートを示す。

第 3 7 図は、本発明の実施例 7 のスプリット機能を説明するための図である。

第 3 8 図は、ミラーディスク変換制御プログラムのフローチャートを示す。

25 第 3 9 図は、本発明の実施例 6 のスプリット制御実行後のデ

ディスクの状態を示す。

発明を実施するための最良の形態

[ 実施例 1 ]

5 第 1 図は、本発明のストレージシステムを用いたシステム全体の構成を説明する図である。ホスト 1 ( 1 5 1 ) 及びホスト 2 ( 1 5 2 ) はホストコンピュータであり、それらの OS のプラットフォームはそれぞれ異なる。この点は後述する。1 2 2 はディスクアレイ制御装置、1 1 7 ~ 1 2 1 はディスクドライ  
10 ブである。本実施例では、ディスクアレイ制御装置 1 2 2 と、ディスクドライブ 1 1 7 ~ 1 2 1 とにより、ストレージサブシステムを構成している。このストレージサブシステムは、S A N ( ストレージエリアネットワーク ) 1 5 3 を介してホスト 1 とホスト 2 とに接続されている。また、ホスト 1 とホスト 2 の  
15 間は L A N ( ローカルエリアネットワーク ) 1 5 0 で接続されている。L A N 1 5 0 は、主にホスト間の通信に利用され、S A N 1 5 3 は、主にストレージサブシステムに対するデータ入出力に利用される。L A N と S A N とでは使用する通信プロト  
20 コルが異なる。一般に、L A N より S A N の方が、データ転送性能が高い。従って、S A N の方が、ストレージサブシステムに対するデータ入出力のような大規模なデータ転送に向く。

ディスクアレイ制御装置 1 2 2 は、ホスト 1 又はホスト 2 から入出力要求が発行されると、その入出力要求を解釈し、その入出力要求に対応するディスクドライブ 1 1 7 ~ 1 2 1 に対し  
25 入出力を行う。ディスクアレイ制御装置 1 2 2 は、ホスト制御

部 1 0 3 , 制 御 プ ロ セ ッ サ 1 0 4 , メ モ リ 1 0 5 , ド ラ イ ブ 制  
御 部 1 1 2 ~ 1 1 6 , お よ び こ れ ら を 接 続 す る バ ス 1 1 1 か ら  
構 成 さ れ る 。 ホ ス ト 制 御 部 1 0 3 は ホ ス ト 1 又 は ホ ス ト 2 か ら  
発 行 さ れ た 入 出 力 要 求 の 受 け 付 け 、 そ の 入 出 力 処 理 が 終 了 す る  
5 と 処 理 終 了 を そ の 要 求 元 の ホ ス ト へ 知 ら せ る 。 ド ラ イ ブ 制 御 部  
1 1 2 ~ 1 1 6 は 、 ド ラ イ ブ 1 1 7 ~ 1 2 1 に 対 す る デ ー タ の  
入 出 力 制 御 を 行 う 。 ホ ス ト 制 御 部 1 0 3 と ド ラ イ ブ 制 御 部 1 1  
2 ~ 1 1 6 は 、 制 御 プ ロ セ ッ サ 1 0 4 か ら の 起 動 / 終 了 , デ ー  
タ 転 送 指 示 に よ り 動 作 す る 。 制 御 プ ロ セ ッ サ 1 0 4 の 動 作 は 、  
10 メ モ リ 1 0 5 内 に 格 納 さ れ た プ ロ グ ラ ム や テ ー ブ ル に 記 述 さ れ  
て い る 。 デ ィ ス ク ア レ イ 読 み 込 み プ ロ グ ラ ム 1 0 6 は 、 ホ ス ト  
1 又 は ホ ス ト 2 か ら 発 行 さ れ た 入 力 要 求 を 制 御 す る 制 御 プ ロ グ  
ラ ム で あ り 、 デ ィ ス ク ア レ イ 書 き 込 み プ ロ グ ラ ム 1 0 7 は ホ ス  
ト 1 又 は ホ ス ト 2 か ら 発 行 さ れ た 出 力 要 求 を 制 御 す る 制 御 プ ロ  
15 グ ラ ム で あ る 。 変 換 プ ロ グ ラ ム 1 0 9 は 、 ド ラ イ ブ 内 に 格 納 さ  
れ た デ ー タ の 形 式 を 変 換 す る プ ロ グ ラ ム で あ る 。 ホ ス ト テ ー ブ  
ル 1 1 0 は 、 接 続 さ れ て い る ホ ス ト が ど の よ う な デ ー タ 形 式 を  
サ ポ ー ト し て い る か を 示 す 情 報 が 格 納 さ れ て い る 。 キ ャ ッ シ ュ  
領 域 1 0 8 は 、 ド ラ イ ブ 1 1 7 ~ 1 2 1 か ら 読 み 込 ま れ た デ ー  
20 タ を 一 時 的 に 格 納 す る 領 域 で あ る 。 ホ ス ト 1 又 は ホ ス ト 2 か ら  
キ ャ ッ シ ュ 領 域 1 0 8 に 格 納 さ れ て い る デ ー タ 、 す な わ ち 、 過  
去 に ア ク セ ス さ れ た デ ー タ の 読 み 出 し が 要 求 さ れ た 場 合 に 、 キ  
ャ ッ シ ュ 領 域 1 0 8 に 格 納 さ れ て い る そ の デ ー タ を ホ ス ト 1 又  
は ホ ス ト 2 に 返 送 し 、 入 出 力 レ ス ポ ン ス の 高 速 化 を 実 現 す る 。  
25 ま た 、 ホ ス ト 1 又 は ホ ス ト 2 か ら 書 き 込 み 要 求 が 発 生 し た 場 合

には、キャッシュ領域 1 0 8 内に書き込みデータを一時的に保持し、キャッシュ領域 1 0 8 へデータが格納された時点で、ホスト 1 又はホスト 2 へ書き込み完了報告を行う。これにより、ホスト 1 又はホスト 2 に対して書き込みを高速に行ったように見せることができる。その後、定期時間毎に起動する書き込み処理（図示せず）等により、キャッシュ領域 1 0 8 からディスクドライブ 1 1 7 ～ 1 2 1 への書き込みが実行される。

第 2 図は、ホスト 1（1 5 1）及びディスクアレイ制御装置 1 2 2 の各階層で、ディスクドライブ 1 1 7 ～ 1 2 1 内のデータがどのように管理されているかを説明するための図である。なお、ホスト 2（1 5 2）についても同様である。

ディスクアレイ制御装置 1 2 2 は、ホスト 1 に対して、複数の物理ドライブを単一の論理ドライブに見せることができる。例えば、2 つのディスクドライブ 1 1 7，1 1 8 を単一の論理ドライブ 2 1 3 に、また、3 つのディスクドライブ 1 1 9，1 2 0，1 2 1 を単一の論理ドライブ 2 1 4 に見せる。すなわち、ホスト 1 5 1 からは、物理ドライブであるディスクドライブ 1 1 7 ～ 1 2 1 ではなく、論理ドライブ 2 1 3，2 1 4 が接続されているように見える。以下の説明では、単にドライブとは、物理ドライブ（2 1 5 ～ 2 1 9）ではなく論理ドライブ（2 1 3，2 1 4）を意味するものとする。

ホスト 1 には、オペレーティングシステム OS 1（2 0 3）と、ファイル 2 1 1 を操作するアプリケーションプログラム 2 0 2 とが搭載されている。OS 1（2 0 3）内には、ファイルを管理するファイルシステム 2 0 4 がある。ファイルシステム

204は、論理ドライブ213および214を、それぞれパーティション領域207及び208と、209及び210とに分割する。アプリケーションプログラム202からは、4つのパーティション領域207、208、209、210が存在しているように見える。アプリケーションプログラム202は、4つのパーティション領域207、208、209、210の各領域に対してファイルの生成、参照又は更新を行う。第2図は、パーティション領域207にファイル211が生成された状態を示している。

10 第3図は、異なるOSのマルチプラットフォームによるファイル共有の概念を説明するための図である。ホスト2(152)上では、OS1とは異なるオペレーティングシステムであるOS2が稼動している。ここで、OS1のファイル管理方法とOS2のそれは異なる。

15 ドライブ213に、ホスト1(151)により生成されたファイルAが格納されていると仮定する。当然のことながら、ホスト1(151)はファイルAを読むことができる。しかし、ホスト2(152)はファイルAを読むことができない。OS1とOS2のファイル管理方法が異なるからである。本発明の  
20 目的は、ホスト1(151)とプラットフォームが異なるホスト2(152)からファイルAにアクセスできるようにすることである。そこで、本発明では、第1図に示したように、ディスクアレイ制御装置122内に変換プログラム109およびホストテーブル110を新たに設ける。これらにより、ホスト2から  
25 ファイルAにアクセスする際、ファイルAの形式をOS2の

形式に変換し、ホスト 1 5 2 からファイル A を読めるようにする。

第 4 図から第 9 図を用いて、本実施例における OS 1 及び OS 2 のファイル管理方法について説明する。

5 第 4 図は、OS 1 のファイルシステム 2 0 4 のディスク領域管理方法を説明するための図である。OS 1 のファイルシステムは、ドライブの領域を複数の固定長ブロック 4 0 1 に区切って管理する。ファイルのデータ長が固定長ブロック 4 0 1 より長い場合には、当該ファイルを複数のブロックに分割して格納  
10 する。その際、これらのブロックが連続した領域に確保されるとは限らない。従って、ファイルシステムは当該ファイルがディスク領域のどこに配置されているかを管理しておく必要がある。それを示す情報をファイル管理情報と呼ぶことにする。太線で示した 2 つの領域は、それぞれパーティション領域を示し  
15 ている。

第 5 図は、OS 2 のファイルシステムのディスク領域管理方法を説明するための図である。OS 2 のファイルシステムは、ドライブの領域を複数の固定長ブロック 5 0 1 に区切って管理する。ファイルのデータ長が固定長ブロック 5 0 1 より長い場  
20 合には、当該ファイルを複数のブロックに分割して格納する。その際、これらのブロックが連続した領域に確保されるとは限らない。ファイルのデータ長が固定長ブロック 4 0 1 より長い場合には、当該ファイルを複数のブロックに分割して格納する。その際、これらのブロックが連続した領域に確保されるとは限  
25 らない。従って、ファイルシステムは当該ファイルがディスク



領域のどこに配置されているかを管理しておく必要がある。OS 2のファイル管理情報はOS 2のファイル管理情報と異なる。また、本実施例では、固定長ブロック501のブロック長は、固定長ブロック401の4倍であると仮定している。なお、太線で示した2つの領域は、それぞれパーティション領域を示している。

第6図は、OS 1のファイルシステムの管理構造を説明するための図である。601から604は第1のパーティション領域、605から608は第2のパーティション領域である。601はブートプログラムが格納されている。ブートプログラムは、ホスト起動時に最初に実行されるプログラムである。602はパーティション情報が格納されており、各パーティションの先頭ブロック等の情報が保持されている。603は、データ領域604に格納されているファイルに関する情報が格納されている。604はデータ領域であり、アプリケーションプログラムが生成したファイルが格納されている。第2のパーティション領域も第1のパーティション領域と同様に構成される。第2のパーティション領域の606には602に格納されているパーティション情報と同じデータが格納されている。これは、パーティション情報はドライブ全体を管理する重要な情報であるため、障害等により第1のパーティション情報602が読み出せなくなった場合、第2のパーティション情報606を使用して処理を継続するためである。

第7図は、OS 2のファイルシステムの管理構造を説明するための図である。701から704は第1のパーティション領

域、705から707は第2のパーティション領域である。ブートプログラム、パーティション情報及びファイル情報の内容、データ領域の役割は、第6図で説明したのと同様である。第7図は、第6図と異なり、第2のパーティション領域にパーティション情報は格納されていない。

第8図は、OS1のファイルシステムとOS2のファイルシステムとの間でファイルを共有する方式を説明するための図である。

800は、OS1のファイル管理情報603（又は607）の内容を示す。801はファイル名が格納される領域である。本実施例ではファイルAというファイル名が格納されている。802には、ファイルAに関するファイル管理情報が格納されている領域の先頭ブロックを示す情報が格納されている。ファイルAのファイル管理情報は、ファイルの所有者が格納されている所有者領域803，ファイルのアクセス許可／不許可を示す情報を格納するファイル属性領域804，ファイルの更新日付が格納されるタイムスタンプ領域805，ファイルサイズを格納するファイルサイズ領域806，ファイルのブロック数を示すブロック数領域807およびファイルAのデータが格納されているディスク領域を示すポインタ（808，809、・・・）から構成される。

810は、OS2のファイル管理情報703（又は706）の内容を示す。811はファイル名が格納される領域である。本実施例ではファイルAというファイル名が格納されている。本実施例ではファイルAというファイル名が格納されている。

領域 8 1 1 に続いて、ファイル A に関するファイル管理情報が格納される。ファイル A のファイル管理情報は、ファイルのアクセス許可／不許可情報を格納するファイル属性領域 8 1 2、ファイルの更新日付が格納されるタイムスタンプ領域 8 1 3、  
5 ファイルサイズを格納するファイルサイズ領域 8 1 4 及びファイル A のデータが格納されている領域を示すポインタ領域 8 1 5、ファイル A が格納されているディスク領域を示すポインタ 8 1 6 から 8 2 4 から構成

8 3 1 及び 8 3 2 は、それぞれポインタ 8 0 8 及び 8 0 9 で  
10 指定されるデータブロックである。これらのデータブロックのブロック長は、ブロック 4 0 1 のブロック長と同一である。また、8 3 3 乃至 8 4 0 は、それぞれポインタ 8 1 6 乃至 8 2 5 で指定されるデータブロックである。これらのデータブロックのブロック長は、ブロック 5 0 1 のブロック長と同一である。  
15 8 3 0 は、ファイル A 全体のデータである。上述したように、OS 1 のファイルシステムがディスク領域を管理する単位であるブロック 4 0 1 のブロック長は、OS 2 のそれであるブロック 5 0 1 のブロック長の 4 倍である。従って、OS 2 は OS 1 の 4 倍のポインタを用いてファイル A のデータを管理すること  
20 になる。例えば、ファイル A が OS 1 上のアプリケーションプログラムで生成されたものである場合、ファイル A を OS 2 から見えるようにするために、ブロック長、すなわち、データ領域の管理単位を変更しなければならない。OS 1 上のアプリケーションプログラムで生成されたファイル A を、OS 2 から見えるようにするためには、OS 1 のファイル管理情報の内容 8  
25

00をOS2のそれである810に変換する処理必要となる。  
その処理については後述する。

第9図は、ホストテーブルの構成を示す図である。ホストIDは、ディスクアレイ制御装置にアクセスする各ホストのIDである。本実施例では、第1図に示すようにディスクアレイ制御装置121にホスト1とホスト2とが接続されているので、それらに対し、それらに対しホストID01、02をそれぞれ割り当てることにする。ホストIDは、ホストがディスクアレイ制御装置に入出力要求を発行する際に、その入出力要求と共に転送されるホストからディスクアレイ制御装置に送信される識別子である。ファイルシステム種別は、各ホストで稼動しているファイルシステムの種類が格納されている。ターゲットファイルシステム種別は、ストレージシステム内に格納されるファイルを生成するファイルシステムの種類が格納されている。

15 ホスト1（ホストID01）については、ファイルシステム種別の欄とターゲットファイルシステム種別欄に記述されているファイルシステムが同じである。これは、ファイルを生成するファイルシステムと、入出力を行いたいファイルシステムが同一であることを意味する。従って、ホスト1は、ストレージシステム内のデータにアクセスすることができる。一方、ホスト2（ホストID02）については、ファイルシステム種別の欄とターゲットファイルシステム種別の欄に記述されているファイルシステムが異なっている。これは、ファイルを生成するファイルシステムと、入出力を行いたいファイルシステムが異なっていることを意味し、従来技術では、OS2はストレージシ

25

システム内のデータにアクセスすることができない。管理単位 1 1 0 4 は、ファイルシステムの管理単位を示している。例えば、OS 1 のファイルシステムの管理単位、すなわち、ブロック 4 0 1 のデータ長は 4 0 9 6 バイトであり、OS 2 のファイルシステム 5 の管理単位、すなわち、ブロック 5 0 1 のデータ長は 1 0 2 4 バイトであることを示している。ホスト ID、ファイルシステム種別、ターゲットファイルシステム種別、管理単位の設定方法にはいくつかの方法が考えられる。例えば、図示していないがディスクアレイ制御装置のコンソールからユーザが設  
10 定する方法、ホストからディスクアレイ制御装置に対して特殊なコマンドを発行し、ホストの情報を転送して設定する方法又は、ファイルシステムのアクセス手順からディスクアレイ制御装置が自動的に判断して設定する方法などがある。

第 1 0 図は変換プログラム 1 0 9 の構成を示す。変換プログラム 1 0 9 は、パーティション変換プログラム 1 0 0 1 とファイルシステム変換プログラム 1 0 0 2 から構成される。  
15

第 1 1 図は、パーティション変換プログラム 1 2 0 2 のフローチャートを示している。パーティション変換プログラム 1 2 0 2 は、OS 1 のディスク管理構造を OS 2 のディスク管理構造 20 に変換する。パーティション変換プログラム 1 2 0 2 は、OS 2 のファイルシステムからでも、OS 1 のファイルシステムにおけるパーティションの存在を認識できるようにするためのプログラムである。

パーティション変換プログラム 1 2 0 2 は、例えばディスク  
25 装置の電源投入時や、ユーザ指示があったとき又は異なる種類

のホスト（異なるファイルシステムが稼動するホスト）から初めてアクセスが発生した場合などに起動される。異なる種類のホストから入出力要求が発生したかどうかは、第10図のホストテーブル内のファイルシステム種別とターゲットファイルシステム種別を比較し、一致していなければ異なる種類のホストからのアクセスと判断することができる。

ステップ1101はオリジナルファイルシステムのパーティション情報読み込みを行う。本実施例では、OS1のファイルシステムのパーティション情報を読み込む。ステップ1102はパーティションの先頭ブロックのブロック番号の変換を行う。OS1とOS2ではディスクの管理単位が異なるので、パーティションの先頭をブロック番号で記述している場合には、同じディスクアドレスでも異なるブロック番号となるからである。ステップ1103は、パーティションのサイズ変換を行う。これもステップ1302と同様の理由により行う処理であり、パーティションの大きさをブロック数で記述している場合には、管理単位が異なると同じパーティション容量でも異なるブロック数となるからである。ステップ1104は、全パーティションを変換したかどうかを判定する。まだ変換していないパーティションが存在する場合、ステップ1101に戻る。このパーティション変換結果（以下「変換後のパーティション情報」と呼ぶ。）は、例えば、第13図に示すようにディスクアレイ制御装置122内のキャッシュ領域108に格納する。変換後のパーティション情報1301をアクセスすれば、異なるファイルシステムからでも、OS1のファイルシステムかにおけるパ

ーティション 1 ( 1 3 0 7 ) 、パーティション 2 ( 1 3 0 8 )  
の存在を認識することができる。

第 1 2 図は、ファイルシステム変換プログラム 1 0 0 2 のフ  
ローチャートを示す。パーティション変換プログラム 1 0 0 1  
5 の実行だけでは、異なるファイルシステムから、ファイルをア  
クセスすることができない。その理由は、OS 1 のファイル管  
理情報 ( 4 0 5 , 4 0 9 ) の形式が OS 2 のそれと異なってい  
るからである。そのため、ファイルシステム変換プログラムは、  
ファイル管理情報を変換する。ファイルシステム変換プログラ  
10 ムは、例えば異なるファイルシステムのホストからアクセスが  
発生した場合に起動される。異なる種類のホストから入出力要  
求が発生したかどうかは、上述のように、第 9 図のホストテー  
ブルを利用すればよい。

ファイル変換システム変換プログラム 1 0 0 2 をステップに  
15 分けて説明する。ステップ 1 2 0 1 は、オリジナルファイルシ  
ステムのファイル管理情報を読み込む。本実施例では、OS 1  
のファイルシステムのファイル管理情報を読み込む。ステップ  
1 2 0 2 は、ファイルの先頭ブロックのブロック番号変換する。  
第 8 図の 8 0 2 の情報から 8 1 5 の情報に変換することを意味  
20 する。これは、第 1 1 図で説明したステップ 1 1 0 2 に関する  
説明と同様の理由によるものである。ステップ 1 2 0 3 は、フ  
ァイルの所有者を変換する。この時、変換後のファイルの所有  
者をファイルの保護等の関係で特定の所有者にしたい場合には、  
あらかじめ変換後の所有者を指示しておけばよい。また、ファ  
25 イルシステムの形式によっては所有者を持たないことがあり、

この場合、本ステップは省略される。ステップ 1 2 0 4 は、ファイルのタイムスタンプを変換する。第 8 図の 8 0 5 のデータを 8 1 3 のデータ形式に変換することを意味する。ステップ 1 5 0 5 は、ファイル属性を変換する。第 8 図の 8 0 4 のデータを 8 1 2 のデータ形式に変換することを示す。この時、変換後のファイル属性を書き込み禁止属性で書き込むことで、元ファイルシステムはファイルを更新できるが、異なるファイルシステムからは更新できないように設定することができる。ステップ 1 2 0 6 は、ファイルのデータが格納されているブロック番号を変換する。第 8 図の 8 1 6、8 1 7、...のデータ形式に変換することを示す。この時、データ領域を指し示すポインタは、OS 1 のファイルシステムが格納したデータ領域を示すように変換される。こうすることで、複数のホストから同じデータを参照することができるようになる。本実施例では、OS 1 のディスク管理単位の大きさが OS 2 のその 4 倍であるので、ステップ 1 2 0 6 の処理により、ポイント 8 0 8 及び 8 0 9 のデータが、ポイント 8 1 6 乃至 8 1 9 及びポインタ 8 2 1 から 8 2 4 の 8 つの領域に格納されるデータに変換される。ステップ 1 2 0 7 は、ファイルの終わりを判定し、1 つのファイルの変換が終了していなければステップ 1 2 0 6 を繰り返す。ステップ 1 2 0 8 は、パーティション内の全ファイル情報を変換したかどうかを判定する。ステップ 1 2 0 9 は、ドライブ内の全てのパーティションについて変換したかどうかを判定する。以上の処理により、異なるファイルシステムへファイル情報の変換が完了する。



第 1 4 図は、ファイルシステム変換プログラム 1 0 0 2 を実行後のファイル管理情報の変換結果は、キャッシュ領域 1 0 8 に、変換後のファイル管理情報 ( 1 4 0 1 , 1 4 0 2 ) として格納される。1 4 0 2 はパーティション 1 ( 1 3 0 7 ) 内のファイル情報を変換した結果であり、1 4 0 2 はパーティション 2 ( 1 3 0 8 ) 内のファイル管理情報を変換した結果である。パーティション変換プログラム 1 0 0 1 が変換した変換後のパーティション情報 1 3 0 1 と、ファイルシステム変換プログラム 1 0 0 2 により変換された変換後のファイル管理情報 ( 1 4 0 1 , 1 4 0 2 ) により、パーティション 1 内のファイル A を、OS 2 に見せることができる。

なお、本実施例では、変換プログラムが 1 種類の OS ( またはファイルシステム ) にしか対応していないが、複数の変換プログラムを追加することにより複数のファイルシステムに対応することができる。具体的には、ホストテーブル 1 1 0 に情報を追加すればよい。これにより、ファイルシステムが変わった場合、ファイルシステムのバージョンアップで管理形式が変更された場合でも、ディスクアレイ制御装置 1 2 2 のコンソールまたは、ホストから変換プログラム 1 0 9 を交換することで容易に対応することができる。

第 1 5 図は、変換プログラム 1 0 9 を実行後のストレージシステムの動作を説明するための図である。本実施例では、ディスクアレイ制御装置 1 2 2 内のキャッシュ領域 1 0 8 に、OS 2 ブートプログラム 1 5 0 1 を格納している。ホスト 1 は、ブートプログラム、パーティション情報及びファイル管理情報に

アクセスするときは、それぞれディスク装置に格納されている  
OS 1 ブートプログラム 4 0 3、パーティション情報 4 0 4 及  
びファイル管理情報 4 0 5 にアクセスする。一方、OS 2 は、  
ブートプログラム、パーティション情報及びファイル管理情報  
5 にアクセスするときは、それぞれキャッシュ領域 1 0 8 に格納  
されている OS 2 ブートプログラム 1 5 0 1、パーティション  
情報 1 3 0 1 及びファイル管理情報 1 4 0 1 にアクセスする。  
しかし、データ領域 4 0 6 内のファイル A には、ホスト 1 から  
もホスト 2 からアクセスし、その内容を見ることができる。  
10 つまり、各ホストからブートプログラム、パーティション情報、  
ファイル管理情報のようなファイルシステム固有のデータへア  
クセスする際には、各ホストのファイルシステムの固有の領域  
にアクセスし、その結果返送されるデータも異なる。一方、デ  
ータ領域内のファイルに関しては、各ホストから見ることがで  
15 きる。

第 1 6 図は、ディスクアレイ読み込みプログラム 1 0 6 のフ  
ローを示す。ステップ 1 6 0 1 は、アクセス要求を発行したホ  
ストに関するホスト ID を取得する。ステップ 1 8 0 2 は、ホ  
ストテーブル 1 1 0 の中から、取得したホスト ID に対応する  
20 ファイルシステム種別、ターゲットファイルシステム種別、管  
理単位を取得する。ステップ 1 6 0 3 は、アクセス要求を発行  
したホストのオリジナルファイルシステムかどうかを判定し、  
オリジナルファイルシステムであればステップ 1 8 1 0 に進み、  
そうでなければステップ 1 8 0 4 に進む。本実施例においては、  
25 ホスト 1 からのアクセスであれば“YES”の方に進み、ホス

ト 2 からのアクセスであれば“NO”の方に進むことになる。

ステップ 1 6 0 4 から 1 6 0 9 は、ディスク装置にファイルを格納したファイルシステムとは異なるファイルシステムから要求があった場合に実行される。ステップ 1 6 0 4 では、ブート

5 プログラムへのアクセスかどうかを判定し、ブートプログラムへのアクセスがあればステップ 1 6 0 7 に進み、そうでなければステップ 1 6 0 5 に進む。ステップ 1 6 0 7 では、変換後のファイルシステムに対応するブートプログラムを読み出す。ステップ 1 6 0 5 は、パーティション情報へのアクセスかどうか

10 を判定し、パーティション情報へのアクセスであればステップ 1 6 0 8 に進み、そうでなければステップ 1 6 0 6 に進む。ステップ 1 6 0 8 では、変換後のパーティション情報を読み出す。ステップ 1 6 0 6 は、ファイル管理情報へのアクセスかどうかを判定し、ファイル管理情報へのアクセスであればステップ 1

15 6 0 9 に進み、そうでなければステップを 1 6 1 0 に進む。ステップ 1 6 0 9 では、変換後のファイル管理情報を読み出す。ステップ 1 6 1 0 は、通常のディスクアレイ読み込み処理を行い、要求されたディスクアドレスに対応するデータを読み出す。ステップ 1 6 1 1 は、読み出したデータをホストへ転送する。

20 これらのステップを実行することにより、異なる OS が稼動する複数のホストからのファイル読み出し要求を処理することができる。

以上、本実施によれば、異なる複数のファイルシステム間でファイルが共有できる。また、変換されたファイルシステムの

25 管理情報は、キャッシュ領域 1 0 8 に配置されているため、デ

ディスク装置の容量を圧迫することがない。なお、ディスク上の全てのファイルを変換しても良いし、特定のファイルのみ変換しても良い。また、異なるパーティションのファイルを、変換時に1つのパーティションにあるように見せることも容易である。

### [実施例 2]

第17図は、本発明の実施例2のディスク上のデータ格納形式を説明するための図である。実施例1では、変換後のパーティション情報1301、変換後のファイル管理情報1401、1402及び、OS2ブートプログラム1501は、キャッシュ領域108に格納されていた。一般に、キャッシュ領域108はバッテリーにてバックアップされているため、ディスクアレイ制御装置122の電源を切った場合でもすぐに内容が失われることはないが、長時間の保存には向かない。変換後のパーティション情報1301及び変換後のファイル管理情報1401、1402は、OS1のファイルシステムのデータから回復可能であるが、それらのデータの回復には時間を要するため、一度変換したそれらの情報は恒久的に保存することが望ましい。

そこで、実施例2では、第17図に示すように、変換データをキャッシュ領域108ではなく、ディスク装置に保存する。ディスク装置には、オリジナルのファイルシステム（本実施例ではOS1のファイルシステム）のブートプログラム403、パーティション情報404、ファイル管理情報405、406と、OS2ブートプログラム1501、変換後のパーティション情

報 1 3 0 1 及び、変換後のファイル管理情報 1 4 0 1 , 1 4 0 2 が格納される。

第 1 8 図は、実施例 2 のディスクアレイ読み込みプログラムのフローを示す。第 1 6 図で説明したフローのステップと共通のステップについては説明を省略し、第 1 6 図のフローと異なるステップのみ説明する。第 1 8 図のフローでは、ステップ 1 9 0 1 が加わる。すなわち、アクセス要求を発行したホストのファイルシステムがオリジナルファイルシステムである場合には、ディスクアドレスにオフセット 1 7 0 3 (第 1 7 図参照) を加算する。ディスク装置内には、OS 2 ブートプログラム 1 5 0 1、変換後のパーティション情報 1 3 0 1 及び、変換後のファイル管理情報 1 4 0 1 , 1 4 0 2 が格納されているからである。ホストからアクセス要求されたディスクアドレスに、オフセット量を加算し、正しいデータをアクセスできるようにする。

なお、OS 2 ブートプログラム 1 5 0 1、変換後のパーティション情報 1 3 0 1 及び、変換後のファイル管理情報 1 4 0 1 , 1 4 0 2 をディスク装置内に格納した場合であっても、アクセス頻度が高いデータはキャッシュ領域 1 0 8 に読み込まれるため、実施例 1 と同等の読み出し性能を得ることができる。また、本実施例 2 では、OS 2 ブートプログラム 1 5 0 1、変換後のパーティション情報 1 3 0 1 及び、変換後のファイル管理情報 1 4 0 1 , 1 4 0 2 をディスク装置内の特定の領域に配置したが、それらのデータの配置は自由に変えられる。従って、ストレージシステムに接続されるホストの OS を変更する場合、新

たなホストを追加接続する場合であっても、オリジナルファイルシステムが管理する空き領域にブートプログラム、変換後のパーティション情報及び、変換後のファイル管理情報を配置すれば、オリジナルファイルシステムのディスク領域構造を変えることなくそれらの変更をすることができる。

### [実施例 3]

実施例 3 では、実施例 1 で使用したホストテーブル 110 を使用しないでディスク共有を行うことを目的とする。第 19 図は、実施例 3 のデータ共有方式を説明するための図である。ホスト 1 (151) とホスト 2 (152) は、SAN 153 を介して、ディスクアレイ制御装置 122 に対し、異なるディスク ID (1903, 1907) を指定してコマンド (1902, 1907) およびデータ (1901, 1905) を転送する。

第 19 図では、ホスト 1 がディスク ID = 3 を指定し、ホスト 2 がディスク ID = 4 を指定する例を示している。ディスクアレイ制御装置 122 は、ディスク ID = 3 を指定してアクセスされた場合は通常のディスク入出力を行い、ディスク ID = 4 を指定してアクセスされた場合は実施例 1 の第 15 図で示した変換後のパーティション情報 1301, 変換後のファイル管理情報 1401, 1402 を利用して入出力処理を行うことを特徴とする。つまり、ディスク ID = 3 はホスト 1 で稼動する OS 1 (デフォルトファイルシステム) 専用であり、ディスク ID = 4 はホスト 2 で稼動する OS 2 専用であることを意味する。

OS の種類毎の制御方法はディスク制御テーブル 1908 のよ

うに記載しておく。

第20図は、実施例3のディスクアレイ読み込みプログラムのフローチャートを示す。第16図で説明したフローのステップと共通のステップについては説明を省略し、第16図のフローと異なるステップのみ説明する。ステップ2001は、ホストが転送したディスクIDを取得する。ディスクIDは、例えばホストがディスクアレイ制御装置に入出力要求を発行した際に、ホストからディスクアレイ制御装置に入出力要求と共に転送される。ステップ2002は、そのホストIDがデフォルトファイルシステムを示すものである場合には、通常ディスクアレイ読み込み処理（ステップ1610）を行い、そうでなければ、前述したステップ1604の処理を行う。

上記処理により、実施例1のホストテーブル110を設定しなくても、異なるファイルシステムでファイルを共有することができる。

本実施例3の変形例として、ディスクアレイ制御装置が複数種類のホスト接続インタフェースを設けている場合、インタフェースの種類に対応してファイルシステムの変換を行うことも可能である。

#### [実施例4]

実施例4は、実施例1ではディスクアレイ制御装置122内で実行されていた変換プログラム109をホスト内で処理することを特徴とする。

第21図は、本発明の実施例4のストレージシステムを用い

たシステム全体の構成を説明する図である。第 2 1 図に示す構成の大部分は第 1 図に示す構成と同じであるが、ホスト 1 ( 1 5 1 ) に変換プログラム 1 0 9 を持たせた点、ディスクアレイ制御装置 1 2 2 に変換制御プログラム 2 1 0 1 を持たせた点で、  
5 第 1 図の構成と異なる。

第 2 2 図は、本実施例におけるファイルシステム変換プログラムのフローチャートを示す。第 1 4 図で説明したステップと同様のステップについては説明を省略する。本フローの特徴は、  
10 ステップ 2 2 0 1 において、変換されたファイル管理情報をホスト 1 からディスクアレイ制御装置 1 2 2 へ特別なコマンドを用いて転送する点である。

ディスクアレイ制御装置 1 2 2 に変換されたファイル管理情報が転送されると、変換制御プログラム 2 1 0 1 が起動される。

第 2 3 図は、変換制御プログラム 2 1 0 1 のフローチャート  
15 を示す。ステップ 2 3 0 1 は、ホスト 1 から転送されたファイル管理情報を受け取る。ステップ 2 3 0 2 では、受け取ったファイル管理情報をキャッシュ領域 1 0 8 へ転送する。以上の処理により、第 1 5 図に示した状態と同様の状態を得ることができる。本実施例では、ファイルシステム変換処理をホスト 1 が  
20 行うため、ディスクアレイ制御装置 1 2 2 内の制御プロセッサ 1 0 4 の負荷が軽くなる。また、ホスト 1 及びホスト 2 上で稼動するのファイルシステムの管理形式が変更された場合でも、ホスト上で実行されるプログラムを交換することにより、その変更柔軟に対応できる。



## [実施例 5]

実施例 5 は、複数のホストから同一ファイルに対して書き込みが発生してもデータの不整合が起こらないようにすることが目的である。

- 5 第 24 図は、本発明の実施例 5 のストレージシステムを用いたシステム全体の構成を説明する図である。第 24 図に示す構成の大部分は第 1 図に示す構成と同じであるが、ディスクアレイ制御装置 122 内に、ロックテーブル 2401 とロック制御プログラム 2402 とを設けた点、及びディスクアレイ書き込みプログラム 2403 が変更されている点で、第 1 図の構成と異なる。ロック制御プログラム 2402 は、複数のホストで共有するファイルに対しあるホストから書き込み要求があった場合に、データの不整合が起こらないように、書き込みを制御するためのプログラムである。ロックテーブル 2401 は、ファイルロックの有無等の情報が格納されている。
- 10
- 15

- 第 25 図は、ロックテーブル 2401 の構成を示す。カラム 2501 は、ストレージシステム内に格納されているファイル名が格納されている。カラム 2502 は、左記のファイルに対し、書き込みを制御するためのロックがかけられているかどうかを示している。カラム 2503 は、左記のファイルにロックを掛けたホストの識別子を示している。カラム 2504 は左記のファイルのロック保持者を示している。カラム 2505 は、左記のファイルにロックをかける回数を示している。
- 20

- 第 26 図は、ホスト 151 内で稼動するアプリケーションが、ディスクアレイ制御装置 122 に対してロックを掛ける / はず
- 25

す制御を説明するための図である。ホスト 1 5 2 についても同様である。ホスト 1 5 1 上では、アプリケーションプログラム 2 0 2 と O S 2 0 3 が稼動しており、O S 2 0 3 内には、ファイルシステム 2 0 4 がある。アプリケーションプログラム 2 0 2 は、ディスクアレイ制御装置 1 2 2 内に格納されたファイルを更新する場合、他のホスト（本実施例ではホスト 2）がそのファイルを同時に更新することを防ぐために、ロックを掛けてからデータの更新を開始する。

すなわち、ホスト 1 5 1 はコマンド：

10 LOCK ("A", "アプリケーション A")

でアクセスしようとするファイル A にロックをかけ、

次にライトコマンド：

WRITE ("A", DATA, 1024, "アプリケーション A")

を送出する。

15 そして、コマンド：

UNLOCK ("A", "アプリケーション A")

によりファイル A に対するロックをはずす。

第 2 7 図は、ロック制御プログラム 2 4 0 4 のフローチャートを示す。ステップ 2 7 0 1 では、ホストからの要求がロックを掛けるコマンドであるかロックをはずすコマンドであるかを判断し、ロックを掛けるコマンドであればステップ 3 0 0 2 の L O C K 処理を実行し（ステップ 2 7 0 2）、そうでなければ U N L O C K 処理を実行する（ステップ 2 7 0 3）。

第 2 8 図は、L O C K 処理のフローチャートを示す。ステップ 2 8 0 1 では、ホストから書き込み要求が発行されたファイ

ルがロックされた状態であるかどうか判断する。これは、ロックテーブル 2401 を参照することで判断可能である。例えば、書き込み要求の対象となってるファイルがファイル A である場合には、現在ロック状態である（第 25 図参照）。そのファイルが既にロックが掛けられた状態であればステップ 2802 に進み、そうでなければステップ 2804 に進む。ステップ 2802 では、そのファイルにロックを掛けたのが自ホストかどうかを判断し、自ホストであればステップ 2803 に進み、そうでなければ 2808 に進み、当該ホストに対しロック失敗を通知する。ステップ 2803 では、そのファイルにロックを掛けたのが同一のロック所有者かどうか判定する。それが同一所有者であればステップ 2804 に進み、そうでなければ 2808 に進み、当該ホストに対しロック失敗を通知する。ステップ 2804 では、ロックテーブル 2401 の対応する欄にホスト ID を登録する。ステップ 2805 では、ロックテーブル 2401 の対応する欄に所有者を登録する。ステップ 2806 では、ロックテーブル 2401 のロック回数に 1 を加える。例えば、ファイル B にロックをかける場合には、ファイル B のロック回数を“1”に更新する。ステップ 2807 では、ホストに対しロック成功を通知する。

第 29 図は、UNLOCK 処理のフローチャートを示す。ステップ 2901 では、ロックテーブル 1401 に登録されているロック回数から 1 を減算する。例えば、ファイル C のロックをはずす場合には、ファイル C のロック回数を“1”に更新する。ステップ 2902 では、ロック回数が 0 以下かどうか判定

し、0以下であればステップ2903に進み、そうでなければ、  
ステップ2906へ進む。ロック回数が0以下であることは、  
すべてのロックが解除されたことを意味する。従って、ステップ  
2903では、ロックテーブル2401のホストIDを削除し、  
5 ステップ2904では、ロックテーブル2754の所有者を削  
除する。ステップ2905では、ロックテーブル2754のロ  
ック回数を0に設定する。ステップ2906では、UNLOCK  
成功をホストへ通知する。

第30図は、実施例3のディスクアレイ書き込みプログラム  
10 2403のフローチャートを示す。ステップ3001は、ロッ  
クがかかっているかどうか検査を行う。このロック検査のフロ  
ーチャートを第31図に示す。

第31図において、ステップ3101は書き込みが発生した  
ディスクアドレスとファイル管理情報からファイル名を検索す  
15 る。ステップ3102は、そのファイル名とロックテーブル2  
401とを参照して、そのファイルがロック済かどうかを判断  
する。ロック済であればステップ3103に進み、そうでなけ  
ればステップ3106へ進む。ステップ3103は、ロックを  
掛けたホストが自ホストと同一か否かを判断し、同一であれば  
20 ステップ3104に進み、そうでなければステップ3105に  
進む。ステップ3105では、所定の時間待ち、再度ステップ  
3102を実行する。ステップ3104では、同一所有者かど  
うかを判定し、同一所有者であればステップ3106に進み使  
用が許可されるが、そうでなければステップ3105に進む。  
25 これによりロックの検査が完了する。

再び、第 3 1 図の説明に戻る。使用許可が得られれば、ステップ 3 0 0 2 に進みホストが転送したディスク I D を取得する。ディスク I D は、例えばホストがディスクアレイ制御装置に入出力要求を発行した際に、ホストからディスクアレイ制御装置  
5 に入出力要求と共に転送される識別子である。ステップ 3 0 0 3 は、ホストテーブル 2 4 0 1 を取得する。ステップ 3 0 0 4 は、アクセス要求を発行したホストのオリジナルファイルシステムかどうかを判定し、オリジナルファイルシステムであればステップ 3 3 1 1 に進み、そうでなければステップ 3 0 0 5 に  
10 進む。オリジナルファイルシステムとは、ディスク装置にファイルを格納したファイルシステムという意味であり、本実施例では、OS 1 のファイルシステムを想定している。ステップ 3 0 0 5 から 3 0 1 0 は、ディスク装置にファイルを格納したファイルシステムとは異なるファイルシステムから要求があった  
15 場合に実行される。すなわち、OS 2 のファイルシステムからアクセス要求があった場合に実行される。ステップ 3 0 0 5 では、OS 2 ブートプログラムへのアクセスかどうかを判定し、OS 2 ブートプログラムへのアクセスであればステップ 3 0 0 8 に進み、そうでなければステップ 3 0 0 6 に進む。ステップ  
20 3 0 0 8 では、OS 2 ブートプログラムにホストから転送されたデータを書き込む。ステップ 3 0 0 6 は、パーティション情報へのアクセスかどうかを判定し、パーティション情報へのアクセスであればステップ 3 0 0 9 に進み、そうでなければステップ 3 0 0 7 に進む。ステップ 3 0 0 9 では、変換後のパーティション情報にホストから転送されたデータを書き込む。ステ  
25

ステップ 3007 は、ファイル管理情報へのアクセスかどうかを判定し、ファイル管理情報へのアクセスであればステップ 3010 に進み、そうでなければステップ 3011 に進む。ステップ 3010 では、変換後のファイル管理情報にホストから転送されたデータを書き込む。ブートプログラム、変換後のパーティション情報、変換後のファイル管理情報の意味は、実施例 1 で説明したのと同様である。ステップ 3011 は、通常のディスクアレイ書き込み処理を行い、要求されたディスクアドレスにデータを書き込む。ステップ 3012 は、書き込み結果をホストへ転送する。上記処理により、複数のホストから同時に書き込みが発生しても、処理を直列化することで、同時書き込みによるデータの不整合を防ぐことができる。

#### [実施例 6]

15 実施例 6 は、複数のホスト 151、152 の間でファイルを共有する場合に、ファイル情報の更新をリアルタイムで行うことを目的とする。一般に、ホスト内で稼動するファイルシステムは、ディスクへの書き込み処理を行う際、その処理の高速化のために、ホスト内のメモリに更新データを一時的に保持しておき、すぐにディスク内のデータを更新をしない。しかし、例  
20 えば、ホスト 151、152 の間でファイルを共有する場合、ホスト 2 はホスト 1 のファイルシステムが、ホスト 1 内のメモリに保持している内容を参照しない。従って、情報の更新が遅れた場合のデータの不整合が生じる可能性がある。本実施例は  
25 この問題を解決することが目的である。

第 3 2 図は、本発明の実施例 6 のストレージシステムを用いたシステム全体の構成を説明する図である。第 3 2 図に示す構成の大部分は第 2 4 図に示す構成と同じであるが、ディスクアレイ制御装置 1 2 2 内に、ホスト 1 ( 1 5 1 ) に、ディスク更新制御プログラム 3 2 0 1 と割り込み制御プログラム 3 2 0 2 を持たせ、ホスト 1 5 1 内の未書き込みデータをディスクアレイ制御装置 1 2 2 に転送する機能を持たせた点、及びディスクアレイ読み込みプログラム 3 2 0 3 が変更されている点で、第 2 4 図の構成と異なる。

10 第 3 3 図は、実施例 6 のディスクアレイ読み込みプログラムのフローチャートを示す。第 1 6 図、第 2 0 図で説明したステップと同様のステップについては説明を省略する。本フローの特徴は、ステップ 3 3 0 1 において、ホストの未書き込みデータをディスクに反映させるための同期制御を行う点である。この同期制御のフローチャートは第 3 4 図に示される。

15 ステップ 3 4 0 1 は、ディスクアレイ制御装置 1 2 2 からホスト 1 5 1 に対して、未書き込みデータのディスク出力要求を発行する。ステップ 3 4 0 2 では、ホスト 1 5 1 から、ディスクへの書き込みが終了するまで待つ。

20 第 3 5 図は、第 3 4 図のステップ 3 4 0 1 が実行された時、ホスト 1 5 1 で実行される割り込み制御プログラム 3 2 0 2 のフローチャートを示す。ステップ 3 4 0 1 が実行されると、ホスト 1 5 1 は、外部割り込みの発生を検出する。ステップ 3 5 0 1 では、この割り込みを受け付ける。ステップ 3 5 0 2 では、25 割り込みの種類がディスクアレイ制御装置 1 2 2 からのディス

ク更新要求かどうかを判定し、ディスク更新要求であればステップ 3503 に進み、そうでなければ他の割り込み判定を行う。ステップ 3503 では、ディスク更新制御プログラム 3201 を起動する。

- 5      第 36 図は、ディスク更新制御プログラム 3201 のフローチャートを示す。ステップ 3601 は、ホスト 151 の未書き込みデータをディスクアレイ制御装置 122 へ書き込む処理を行う。これは、ホスト 151 で稼動している OS 1 はディスク更新機能を持っており、OS 1 がこの機能呼び出すことで実現する。
- 10    ステップ 3602 は、ディスクアレイ制御装置 122 に対して未書き込みデータの更新が完了したことを通知する。上記処理により、複数のホストからディスクを共有する場合に、ホストの更新結果を、もう一方のホストのアクセスに即座に反映させることができる。

15

#### [実施例 7]

- 実施例 7 は、複数のホスト間でファイルを共有する場合に、その共有による性能低下を改善することが目的である。すなわち、複数のホストが単一のディスクにアクセスすることで競合が発生し性能が低下することがある。本実施例は、ディスク・ミラーリングにおけるスプリット機能を用いて、ファイル共有時でも性能が低下しない共有方法を提供する。
- 20

- 第 37 図は、ディスクアレイのスプリット機能を説明するための図である。ディスクアレイ制御装置 122 は、入出力性能と信頼性を向上させるためにディスク 1 (3705) とディス
- 25



ク 2 ( 3 7 0 6 ) に同一データを書き込む (ミラーリング) 機能を有する。スプリット機能とは、スプリット制御とマージ制御からなる。スプリット制御 3 7 0 2 は、そのミラー化された 2 つのディスクを一時的に 2 つの独立したディスクとしてホストにアクセスさせる。マージ制御 3 7 0 3 は、ホストの処理が完了した時点で 2 つに分けたディスクを 1 つに戻すと共に、2 つに分けられていた間に更新されたデータを互いのディスクに反映させる。本実施例では、これらの機能に加えて、ミラーディスク変換制御プログラム 4 2 0 4 を設け、スプリット機能をファイル共有に適用する。このミラー化によって、2 つのディスク装置 3 7 0 5 , 3 7 0 6 の内容は同一のものとなっている。個々のディスク内のデータ構造は第 1 7 図のそれと同じである。

第 3 8 図は、ミラーディスク変換制御プログラム 3 7 0 4 のフローチャートを示す。第 1 2 図で説明したステップと同様のステップについては説明を省略する。本フローの特徴は、ステップ 3 8 0 1 の同期制御、ステップ 3 8 0 2 のスプリット制御及びステップ 3 8 0 3 の変換結果をディスク 2 に反映する点である。ステップ 3 8 0 2 は、スプリット制御プログラム 3 7 0 2 を起動し、ミラー化されたディスク 1 ( 3 7 0 5 ) とディスク 2 ( 3 7 0 6 ) とを分ける。ステップ 3 8 0 3 では、分けられたいずれか一方のディスクに変換したファイル情報を格納する。すなわち、実施例 1 で説明したように、OS 2 ブートプログラム 1 5 0 1、変換後のパーティション情報 1 3 0 1、変換後のファイルシステム情報 1 4 0 1 , 1 4 0 2 を一方のディスクに格納する。以上の処理により、ミラー化されたディスクの

一方は、オリジナルファイルシステムとは異なるファイルシステムに関する情報が格納されたディスクになる。第39図は、本実施例のスプリット制御実行後のディスクの状態を示す。ディスク1(3701)はOS1の入出力専用ディスクとして使用し、ディスク2(3702)はOS2に入出力専用のディスクとして使用することで、複数のホストからディスクを共有する場合でも、異なるディスクに分散してアクセスすることになるため、競合による性能低下を押さえることができる。一方のディスク内のデータが更新された場合には、ステップ3803により、その更新内容を反映させる。

なお、ミラーディスク変換制御プログラムは、ファイル共有を開始したいときに、コンソールやホストから起動させてもよいし、また異なるプラットフォームのホストから始めてアクセスがあった時等に起動させてもよい。

15 以上説明したように、本発明よれば、複数の異なるプラットフォームのホスト間で共有するディスク装置内に格納されたファイルを各プラットフォームのホスト間で共有することが可能となる。

## 請 求 の 範 囲

1. 第 1 の形式によりファイルの格納場所情報を管理する第 1 のオペレーティングシステムが稼動する第 1 のホストコンピュータと、上記第 1 の形式とは異なる第 2 の形式によりファイルの格納場所情報を管理する第 2 のオペレーティングシステムが稼動する第 2 のホストコンピュータとに接続されたストレージシステム内に格納されているファイルであって、上記第 1 の形式によりその格納場所情報が管理されている上記ファイルを、  
10 上記第 2 のホストコンピュータから読み出すためのファイル共有方法であって、

上記第 1 の形式による上記ファイルの格納場所情報を上記第 2 の形式による格納場所情報に変換し、

- 上記第 2 の形式の格納場所情報に基づき上記ファイルを読み  
15 出すステップを有するファイル共有方法。

2. 請求項 1 に記載のファイル共有方法であって、

- 上記第 1 のオペレーティングシステムは第 1 の固定長のブロック単位でファイルの格納場所を管理しており、上記第 2 のオペレーティングシステムは上記第 1 の固定長とは異なる第 2 の  
20 固定長のブロック単位でファイルの格納場所を管理しており、

上記変換するステップは、上記第 1 の固定長のデータ長と上記第 2 の固定長のデータ長との比に基づき、第 1 の形式による上記ファイルの格納場所情報を上記第 2 形式による格納場所情報に変換するファイル共有方法。

- 25 3. 請求項 1 に記載のファイル共有方法であって、

上記変換するステップで変換した上記第2の形式による格納場所情報を上記ストレージシステム内のキャッシュ領域に格納するファイル共有方法。

4. 第1の形式によりファイルの格納場所情報を管理する第1  
5 のオペレーティングシステムが稼動する第1のホストコンピュータと、上記第1の形式とは異なる第2の形式によりファイルの格納場所情報を管理する第2のオペレーティングシステムが稼動する第2のホストコンピュータとに接続されるストレージシステムであって、
- 10 データを格納する複数のディスクドライブと、  
上記第1のホストコンピュータと上記第2のコンピュータと接続するためのインターフェースと、上記ディスクドライブと接続するためのインターフェースとを有するディスク制御装置とを有し、
- 15 上記ディスク制御装置は、  
上記複数のディスクドライブの何れかに格納されているファイルの上記第1の形式による格納場所情報に対応する上記第2の形式による格納場所情報を保持する手段と、  
上記第2のホストコンピュータから上記ファイルにアクセス  
20 要求があった場合、上記第2の形式による格納場所情報に基づき上記ファイルを読み出す手段とを有するストレージシステム。
5. 請求項4に記載のストレージシステムであって、  
上記制御装置は、  
上記第1のオペレーティングシステムがファイルの格納場所  
25 を管理する際に用いる固定長ブロックのデータ長と、上記第2

のオペレーティングシステムがファイルの格納場所を管理する際に用いる固定長ブロックのデータ長との比に応じて、上記第1の形式による上記ファイルの格納場所情報を上記第2形式による格納場所情報に変換する手段を有するストレージシステム。

5 6. 請求項4に記載のストレージシステムであって、

上記ファイルの上記第1の形式による格納場所情報に対応する上記第2の形式による格納場所情報を、上記ディスク制御装置の代わりに上記複数のディスクドライブの内の何れかが保持するストレージシステム。

10 7. 請求項6に記載のストレージシステムであって、

上記複数のディスクドライブの内の第1のディスクドライブと第2のディスクドライブは、それぞれ、上記ファイルの上記第1の形式による格納場所情報と、それに対応する上記第2の形式による格納場所情報と保持し、上記第1のディスクドライブと上記第2のディスクドライブの双方は上記ファイルを保持するストレージシステム。

8. 請求項7に記載のストレージシステムであって、

上記ディスク制御装置は、上記第1のホストから上記ファイルへアクセス要求があった場合、上記第1のドライブにアクセスし、上記第2のホストから上記ファイルへのアクセス要求があった場合には上記第2のドライブにアクセスするストレージシステム。

9. 請求項4に記載のストレージシステムであって、

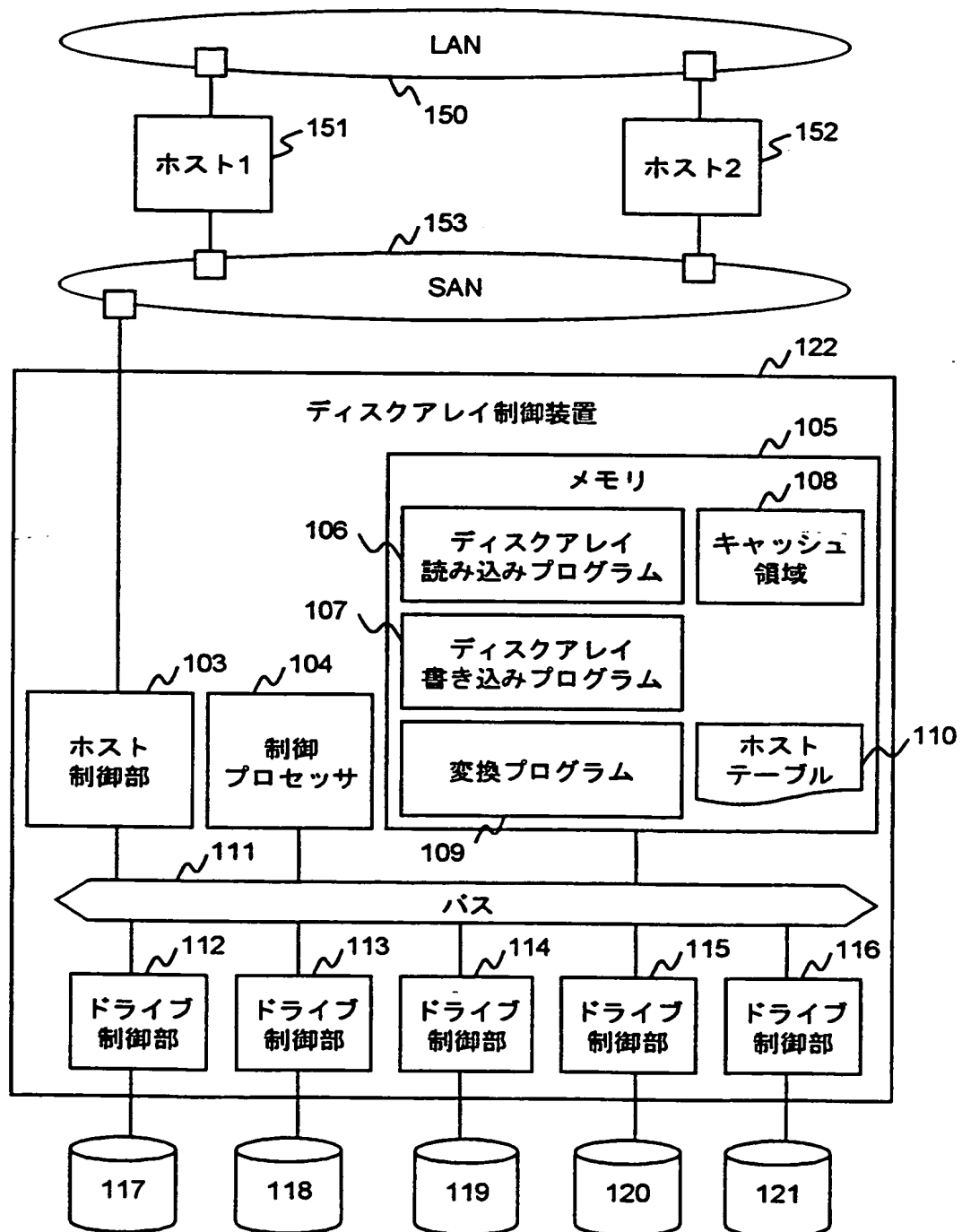
上記制御装置は、

25 上記第1のホストコンピュータからの上記ファイルへのアク

セス要求と上記第２のホストコンピュータからの上記ファイルへのアクセス要求とが競合した場合に、上記第２のホストコンピュータからのアクセス要求を排他制御する手段を有するストレージシステム。

1 / 39

第 1 図

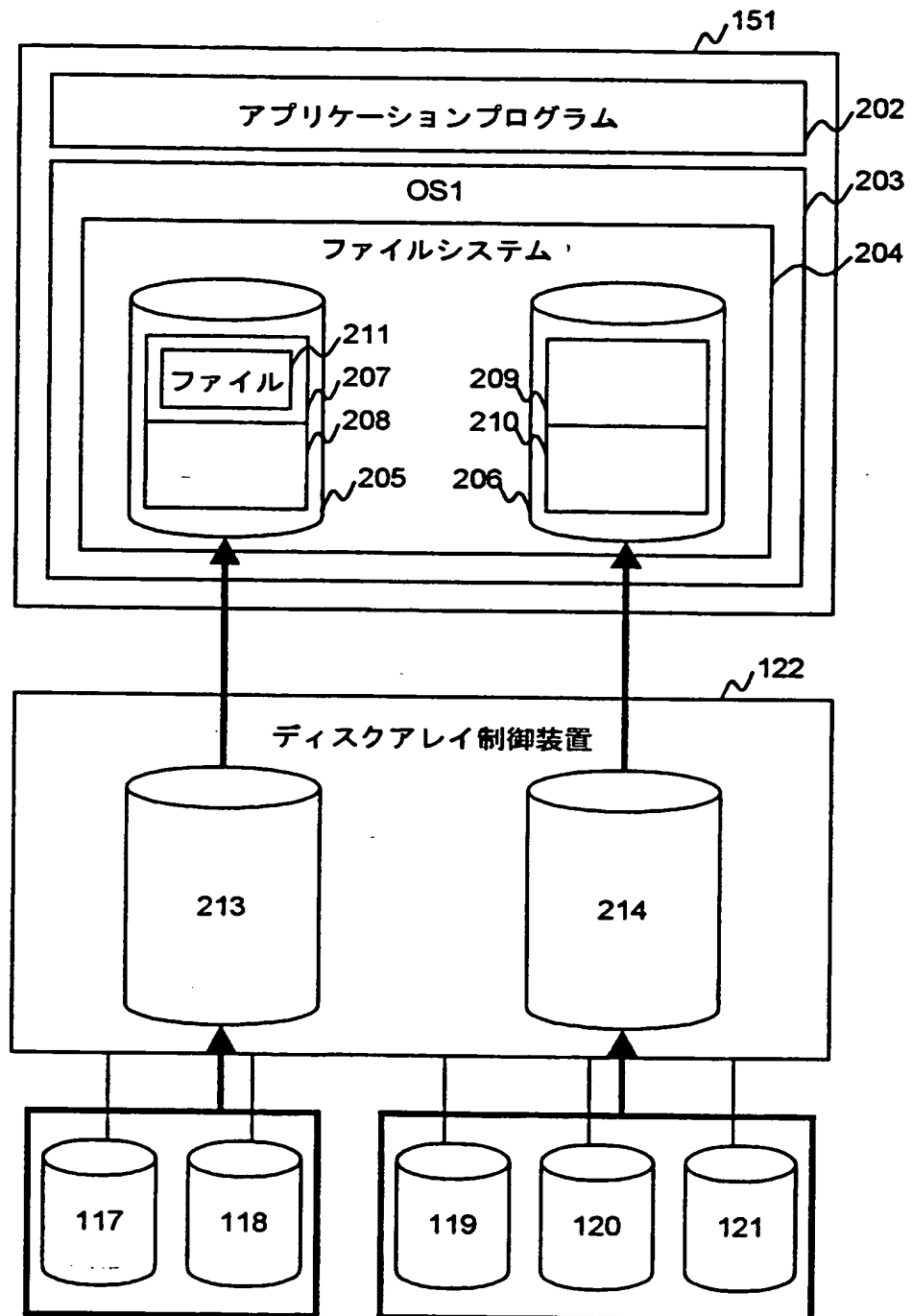


**THIS PAGE BLANK (USPTO)**



2 / 39

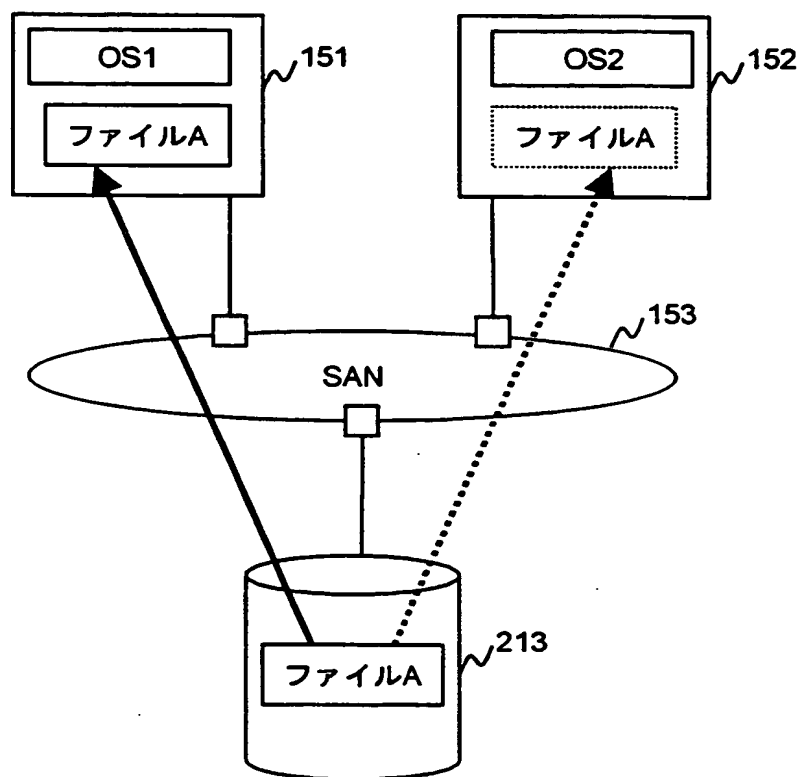
第 2 図



**THIS PAGE BLANK (USPTO)**

3 / 39

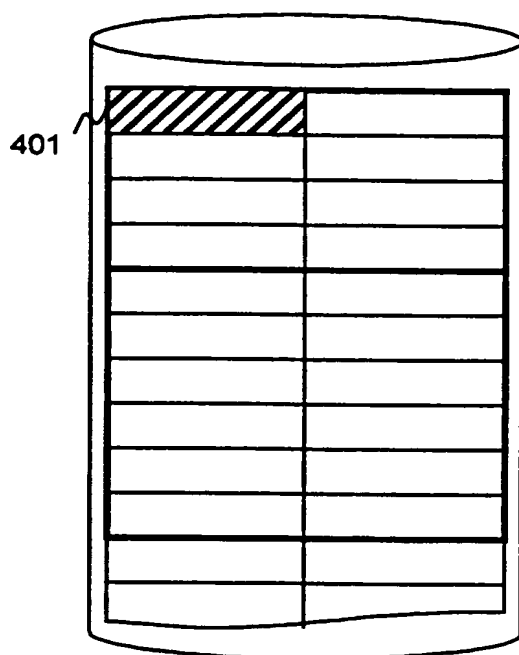
第3図



**THIS PAGE BLANK (USP16)**

4 / 39

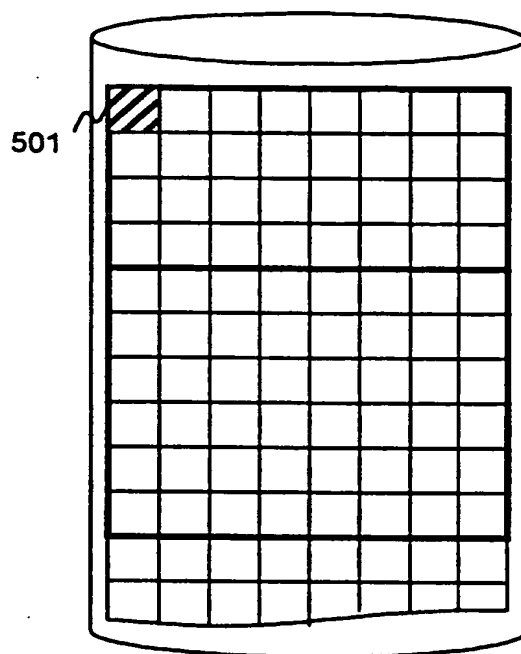
## 第 4 図



**THIS PAGE BLANK (USPTO)**

5/39

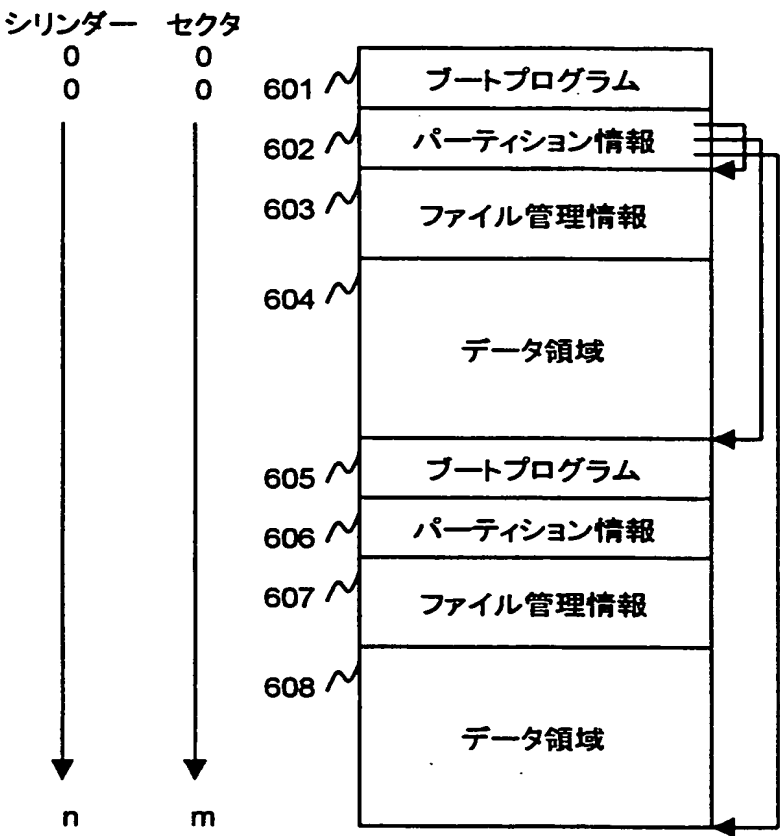
第5図



**THIS PAGE BLANK (USPTO,**



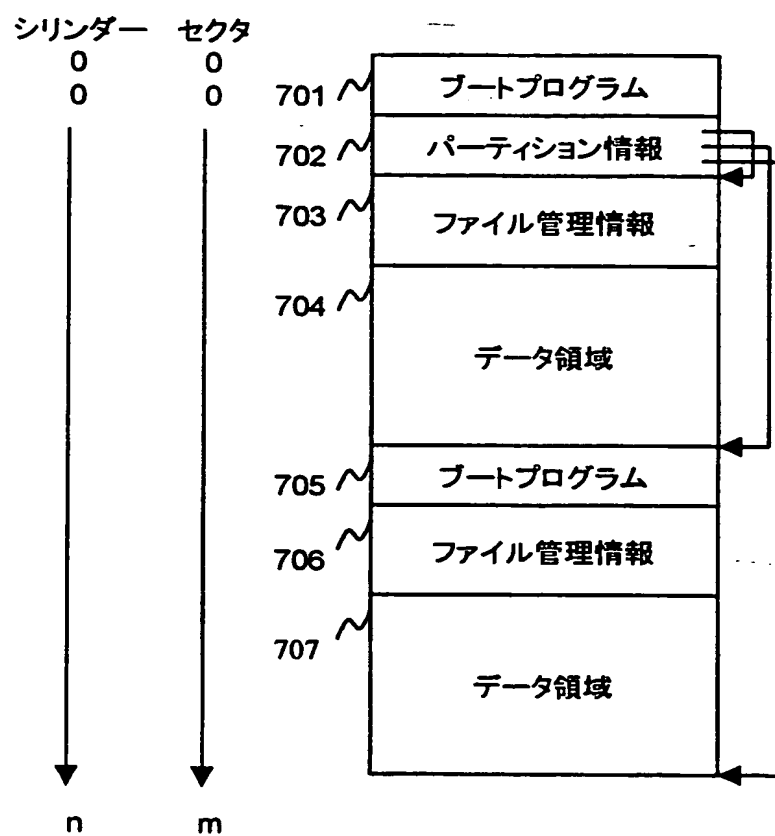
第 6 図



**THIS PAGE BLANK (USPTO)**

7/39

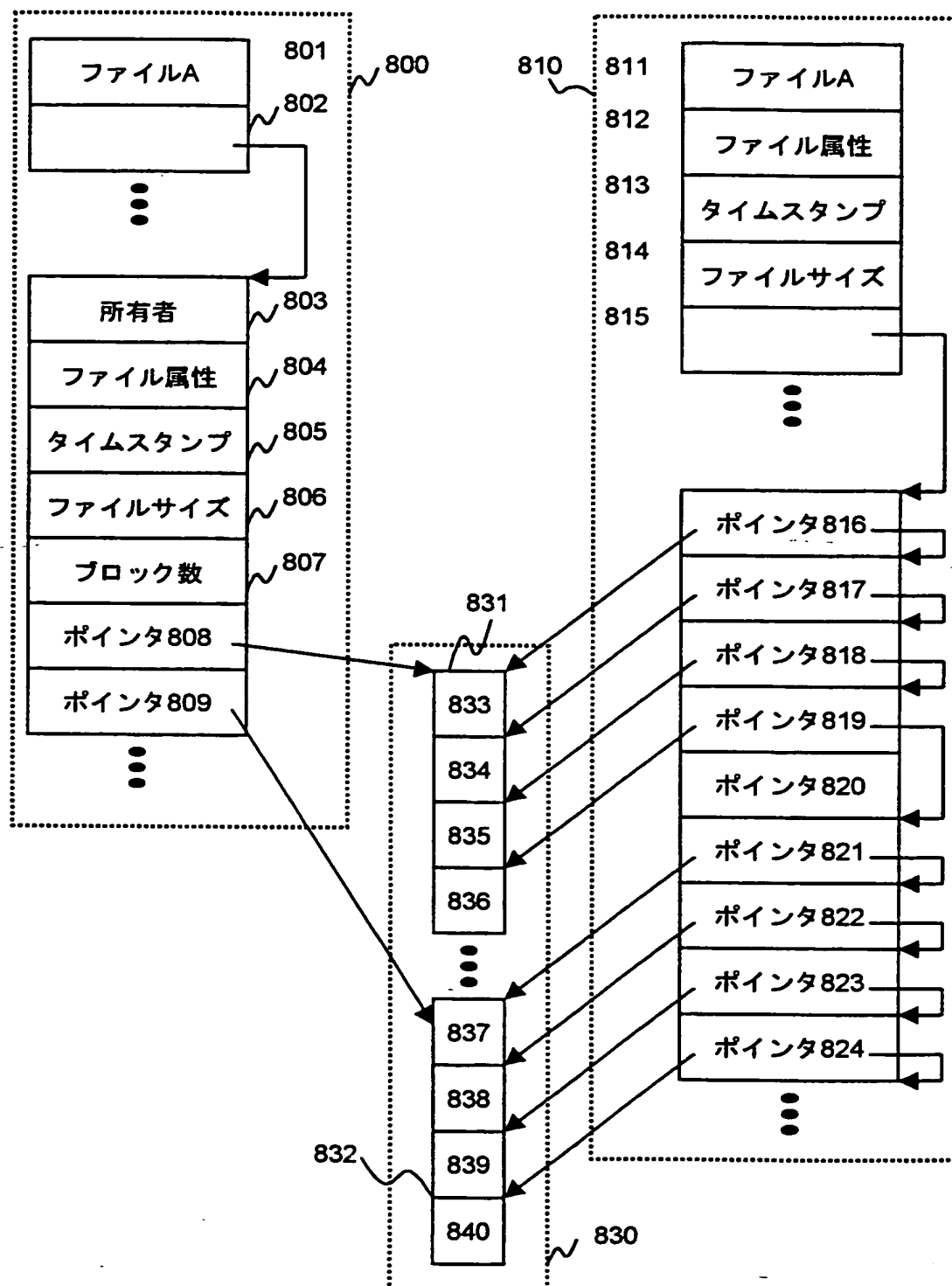
第7図



**THIS PAGE BLANK (USPTO)**

8/39

第8図



**THIS PAGE BLANK (USPTO)**

9 / 39

## 第 9 図

~ 110

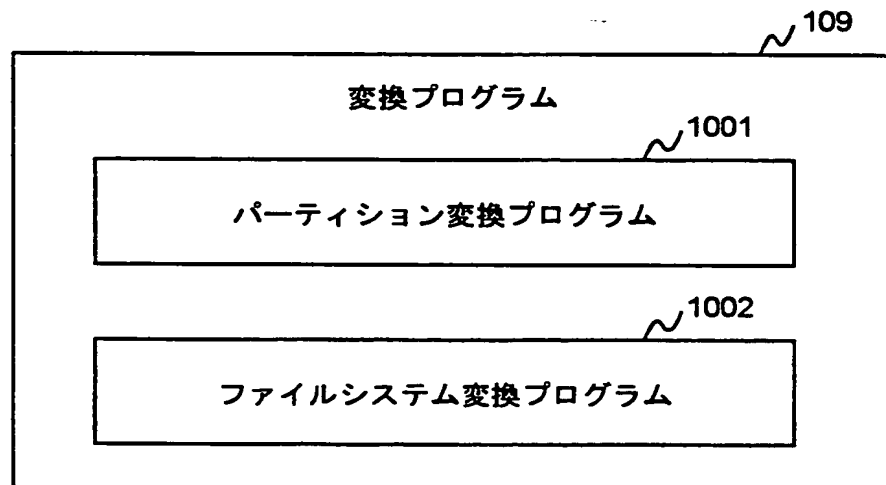
ホストID	ファイルシステム 種別	ターゲット ファイルシステム種別	管理単位
01	ファイルシステム1	ファイルシステム1	4096
02	ファイルシステム2	ファイルシステム2	1024
⋮	⋮	⋮	⋮

**THIS PAGE BLANK (USPTO)**



10 / 39

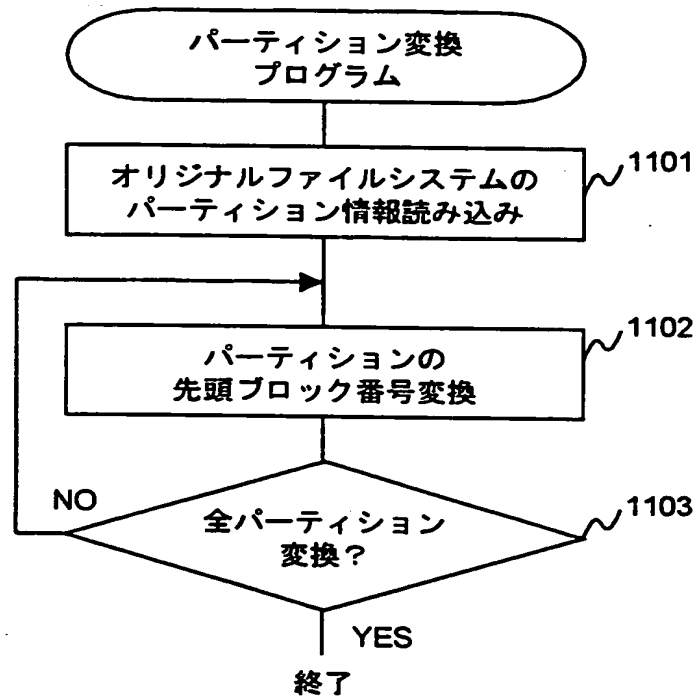
第 10 図



**THIS PAGE BLANK (USPTO)**

11 / 39

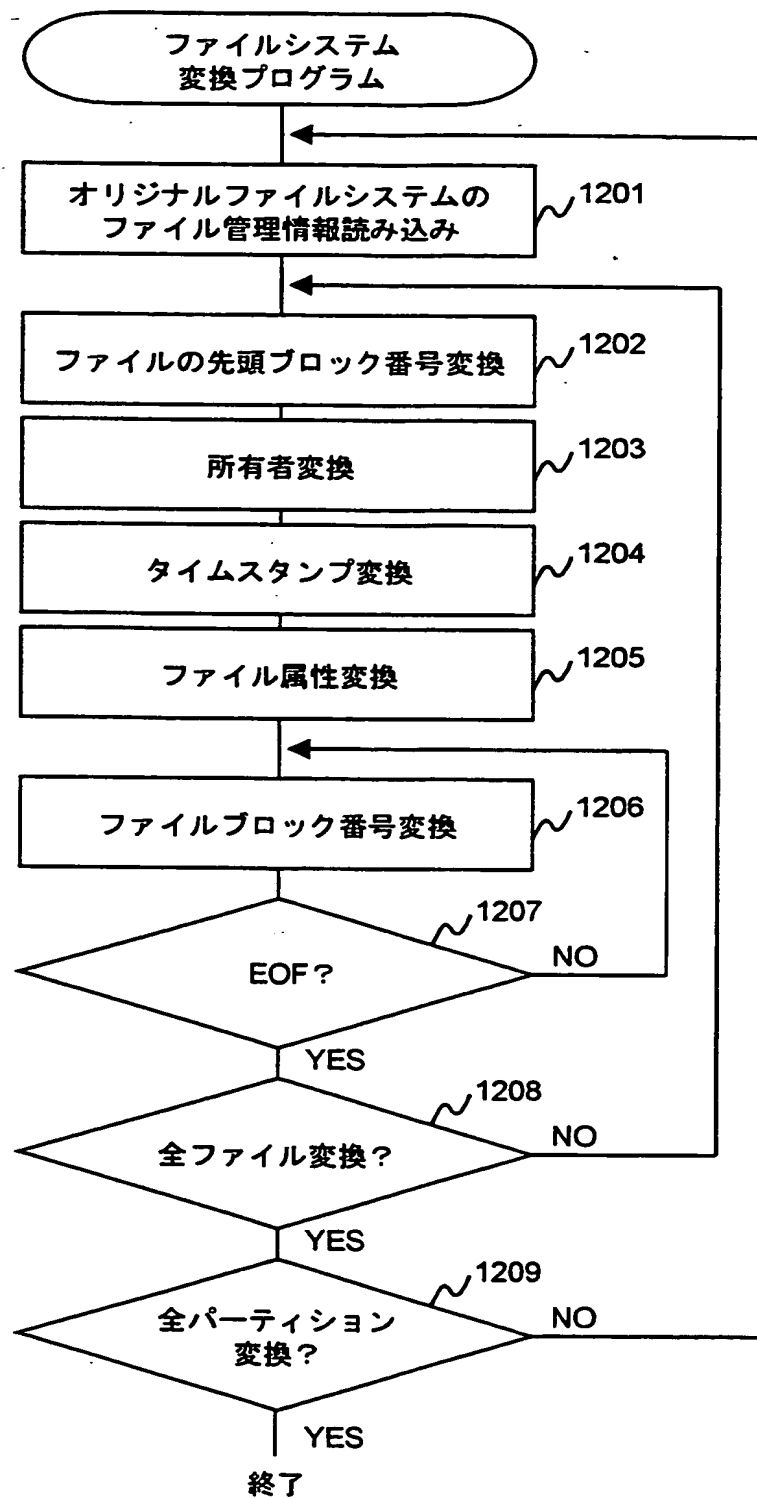
## 第 1 1 図



**THIS PAGE BLANK (USPTO)**

12 / 39

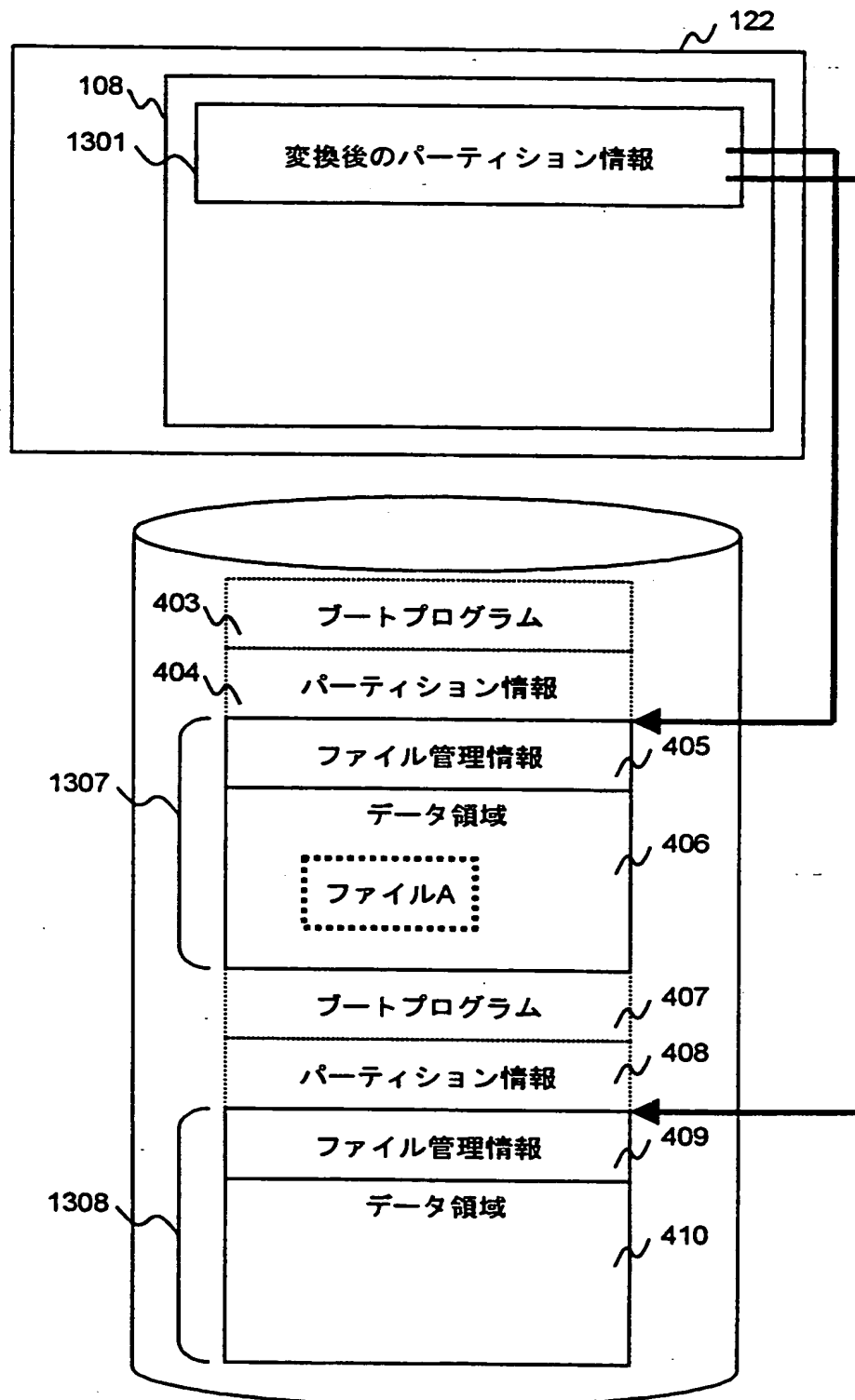
## 第 1 2 図



**THIS PAGE BLANK (USPTO)**

13 / 39

## 第 1 3 図

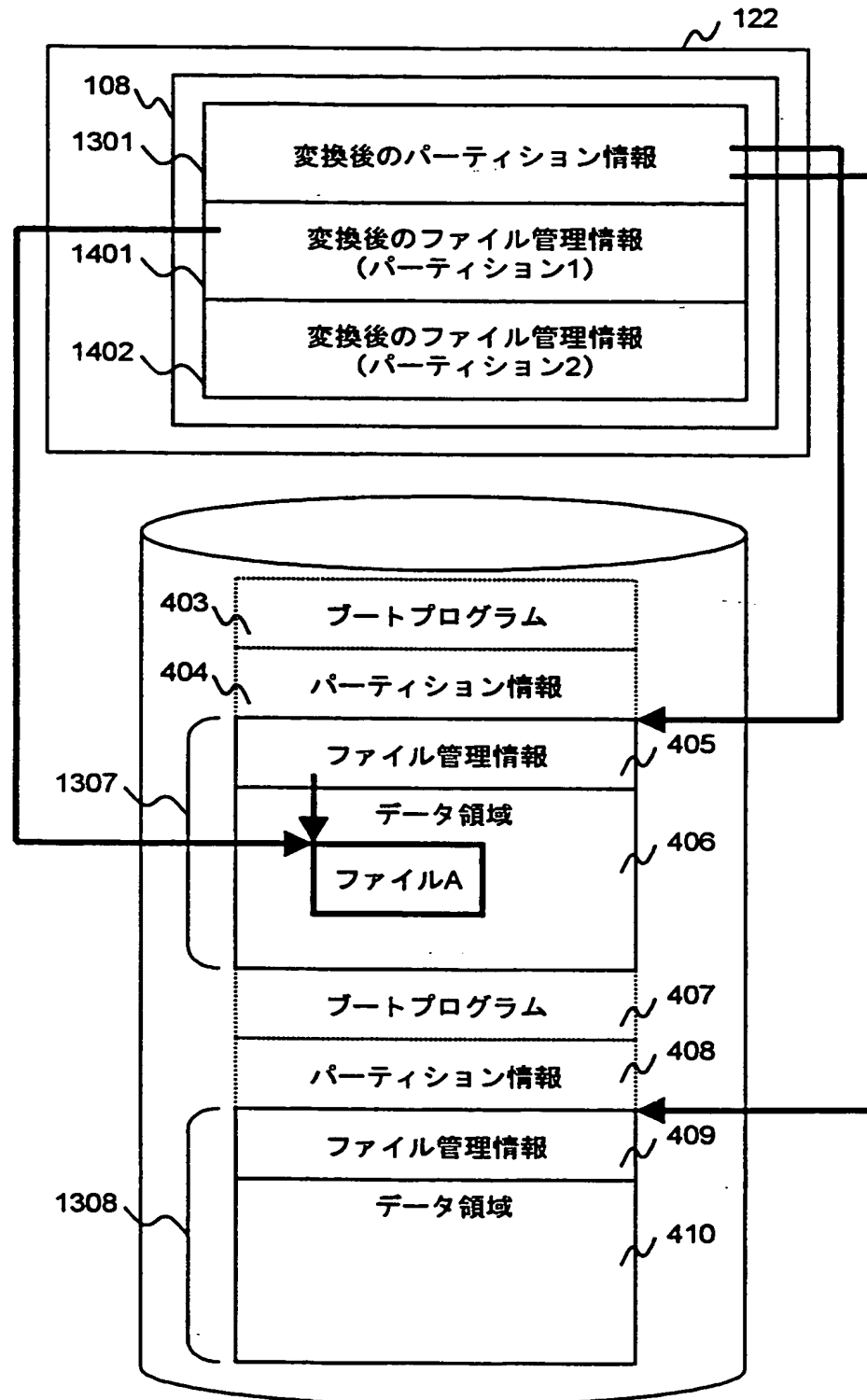


**THIS PAGE BLANK (USPTO)**



14 / 39

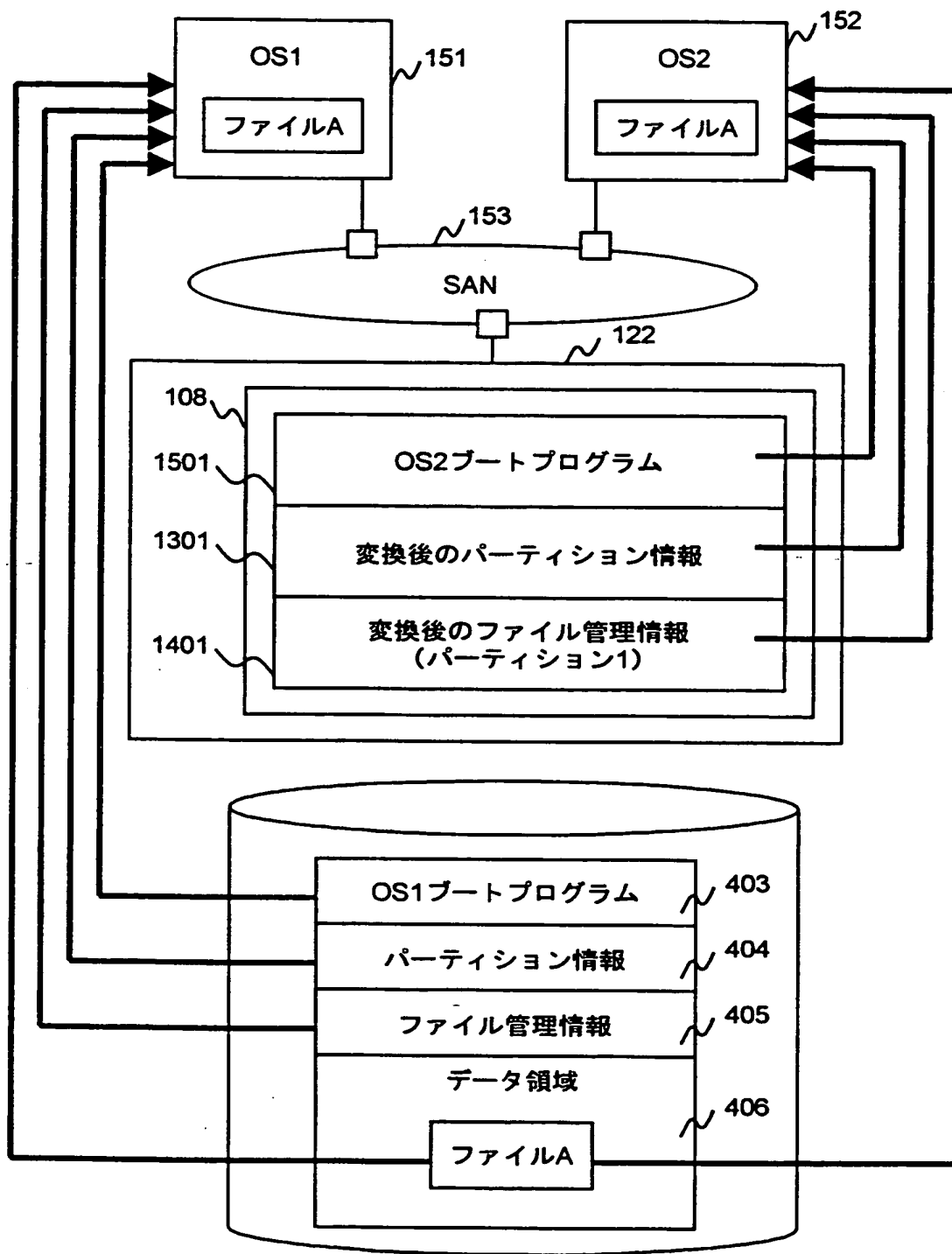
## 第 1 4 図



**THIS PAGE BLANK (USPTO)**

15 / 39

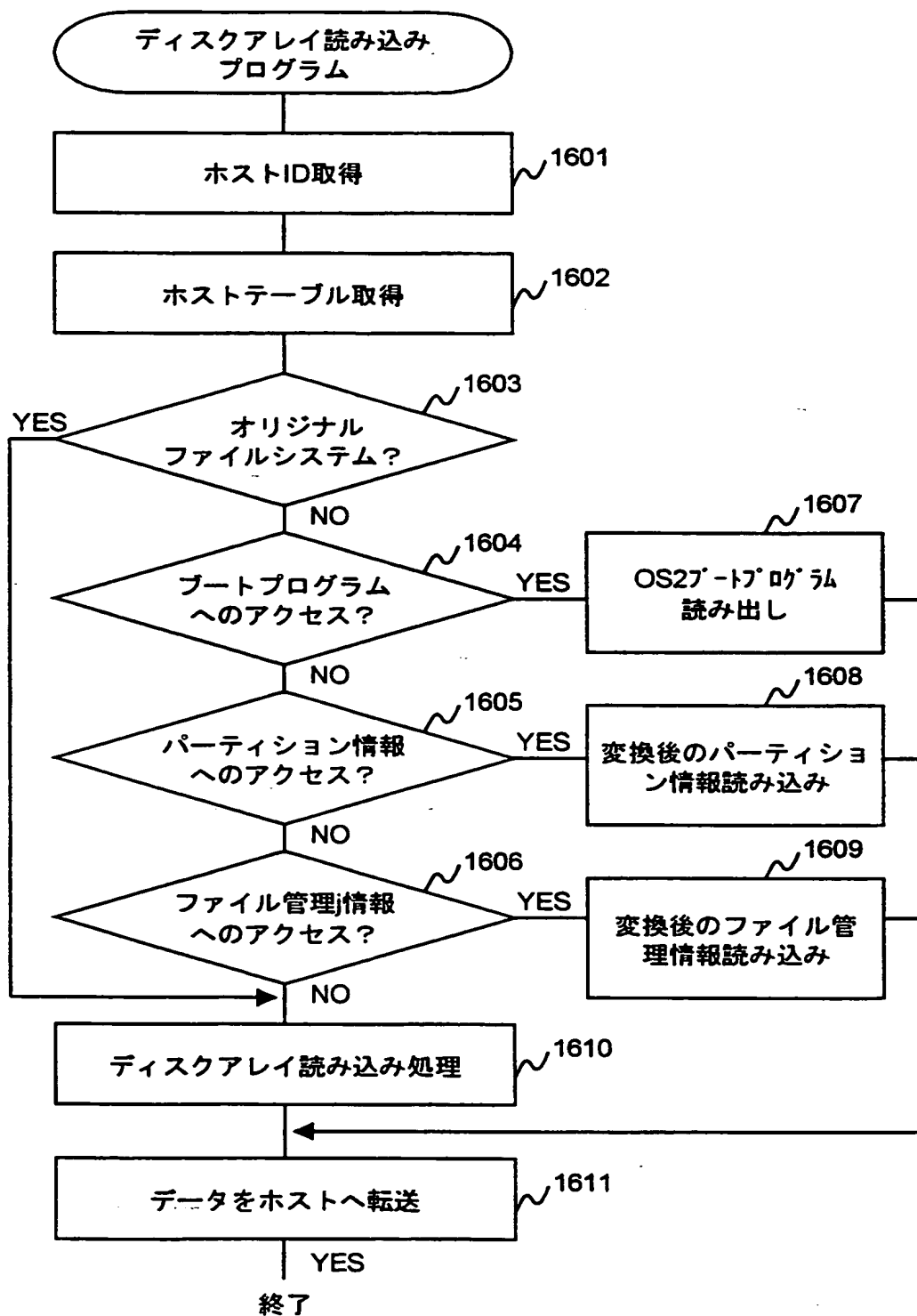
第 1 5 図



**THIS PAGE BLANK (USPTO)**

16 / 39

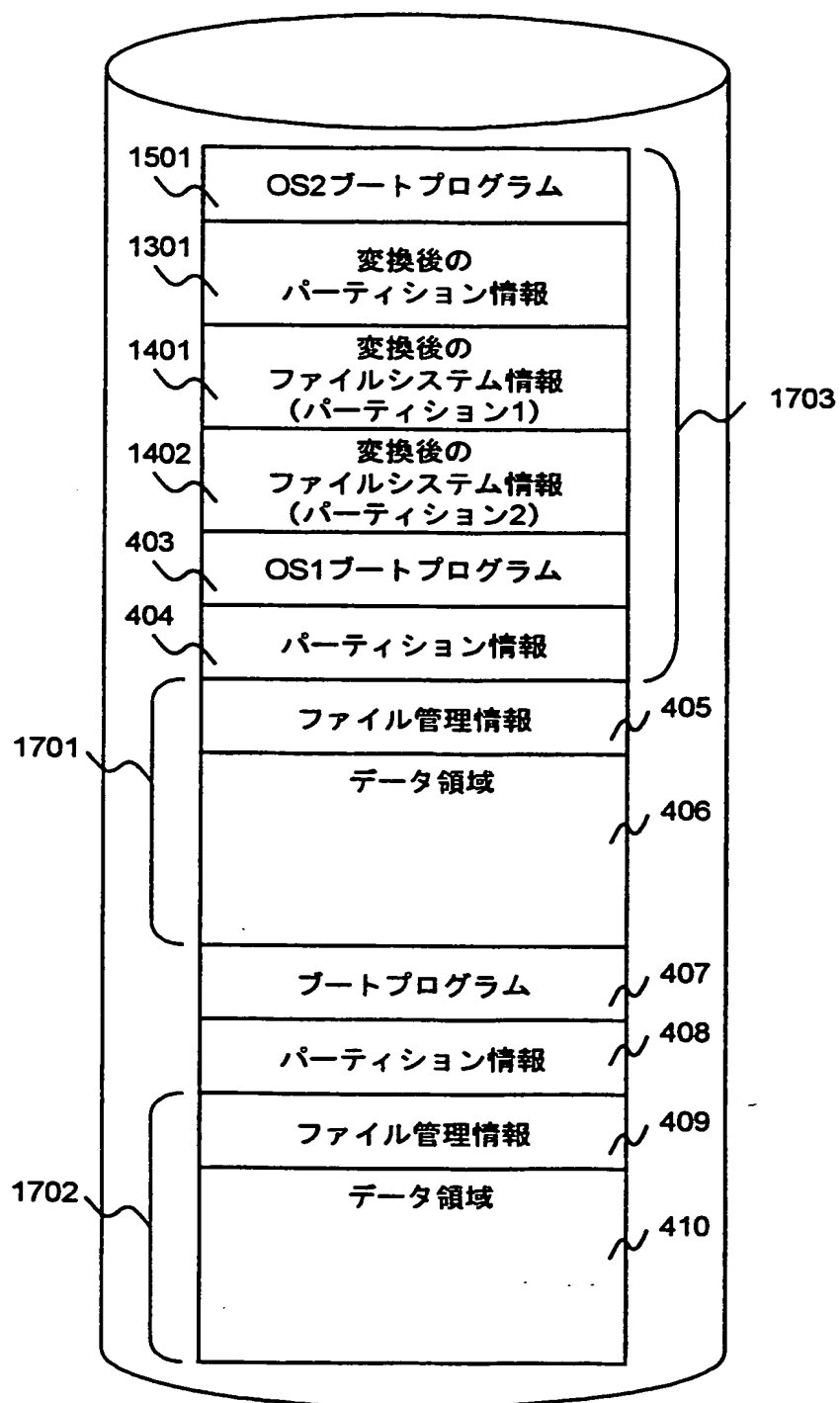
## 第 1 6 図



**THIS PAGE BLANK (USPTO)**

17/39

第17図

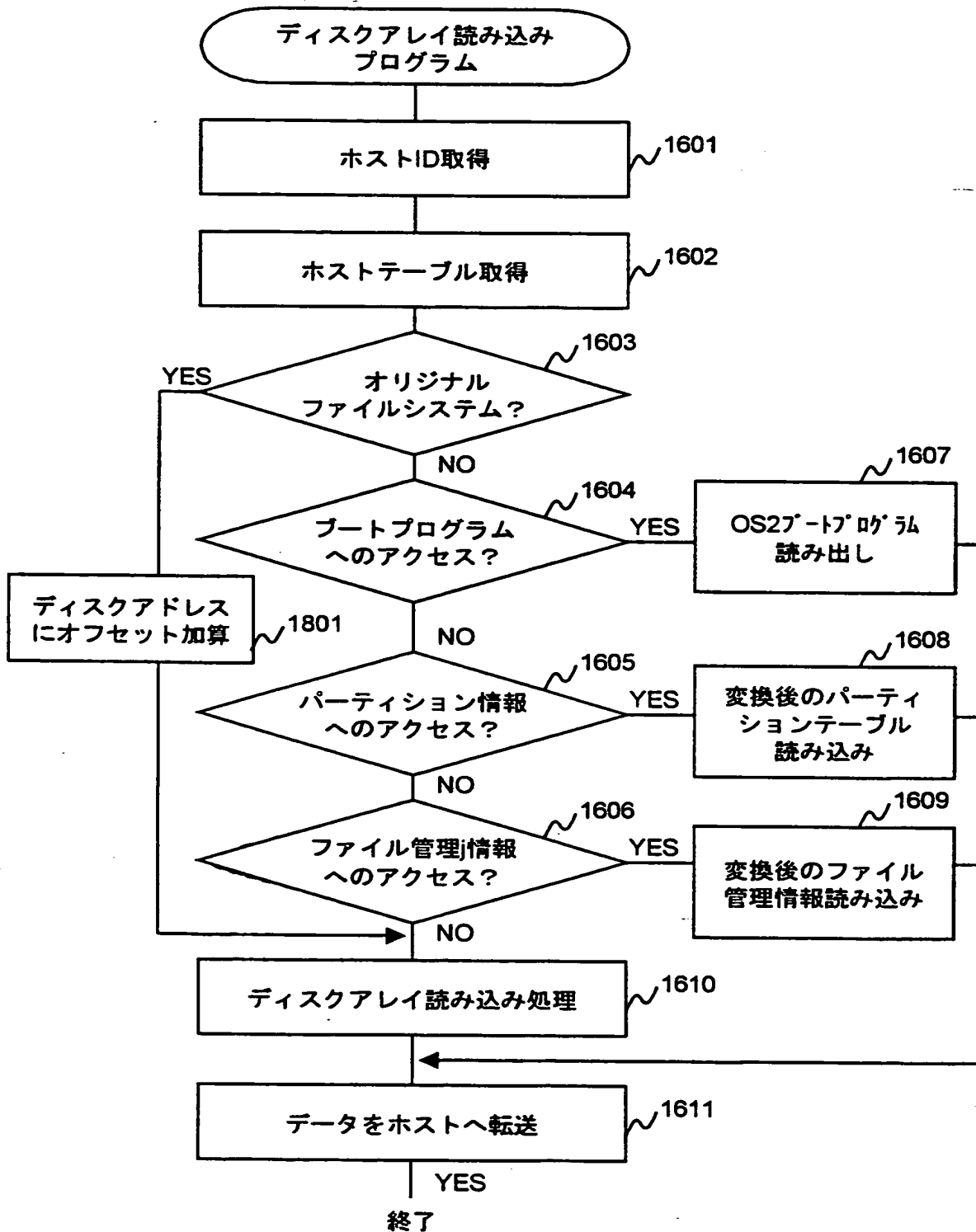


**THIS PAGE BLANK (USPTO)**



18 / 39

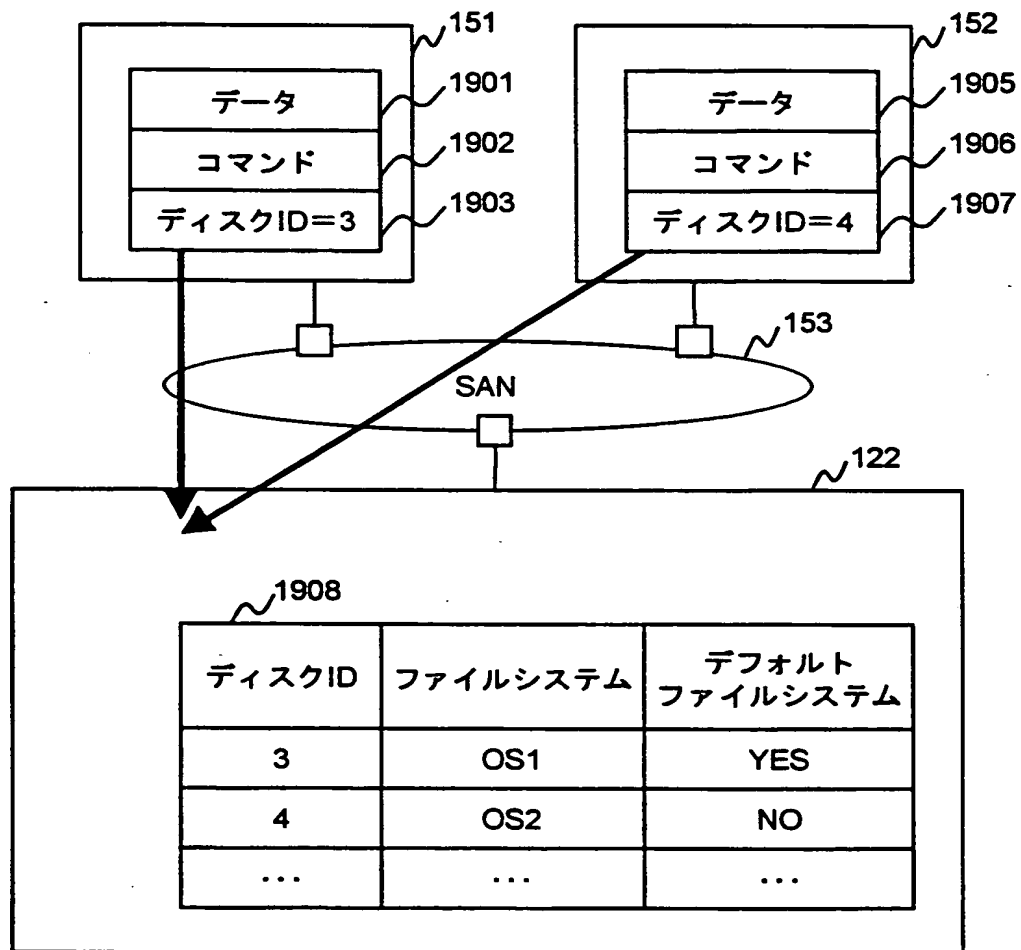
第 18 図



**THIS PAGE BLANK (USPTO)**

19/39

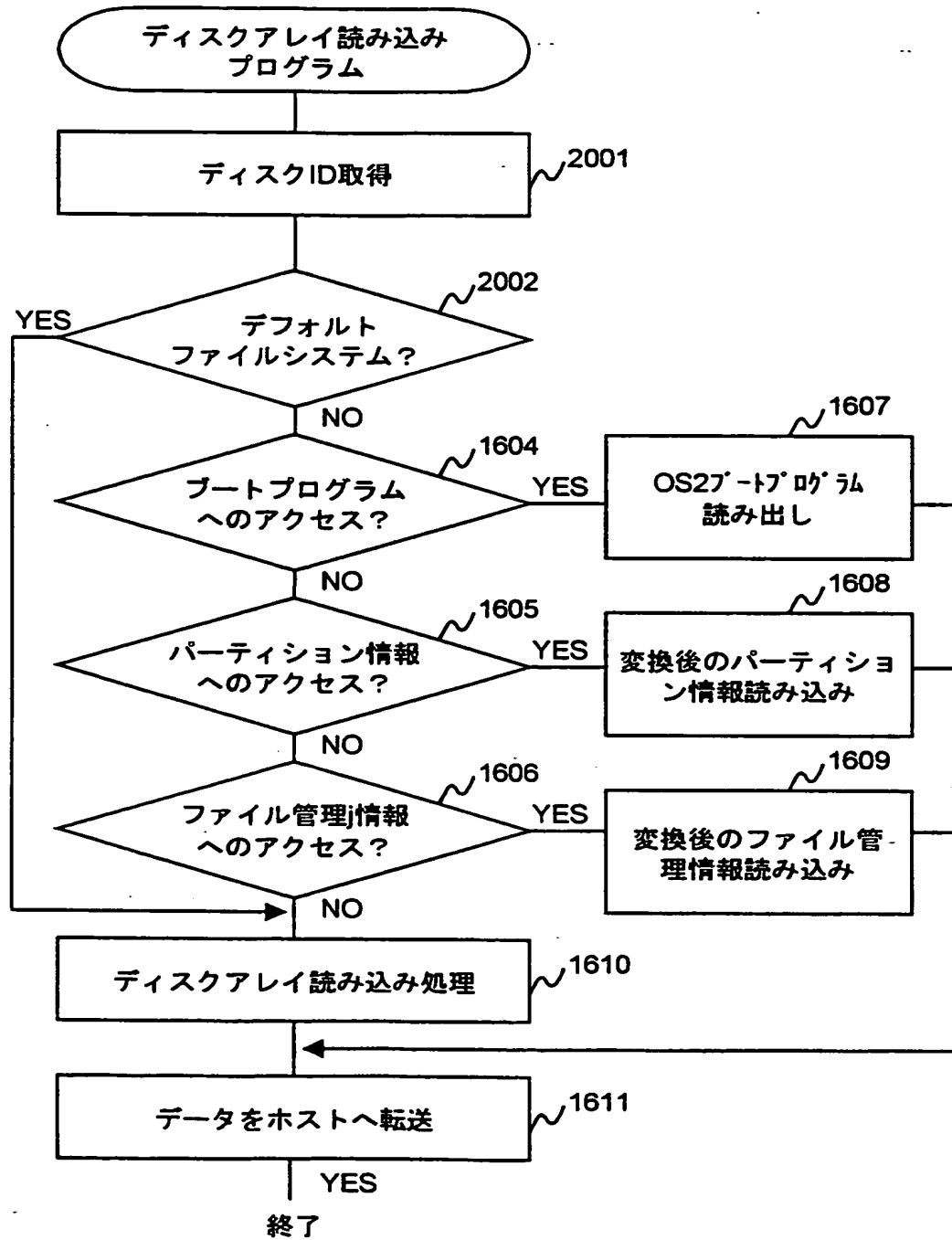
第19図



**THIS PAGE BLANK (USPTO)**

20 / 39

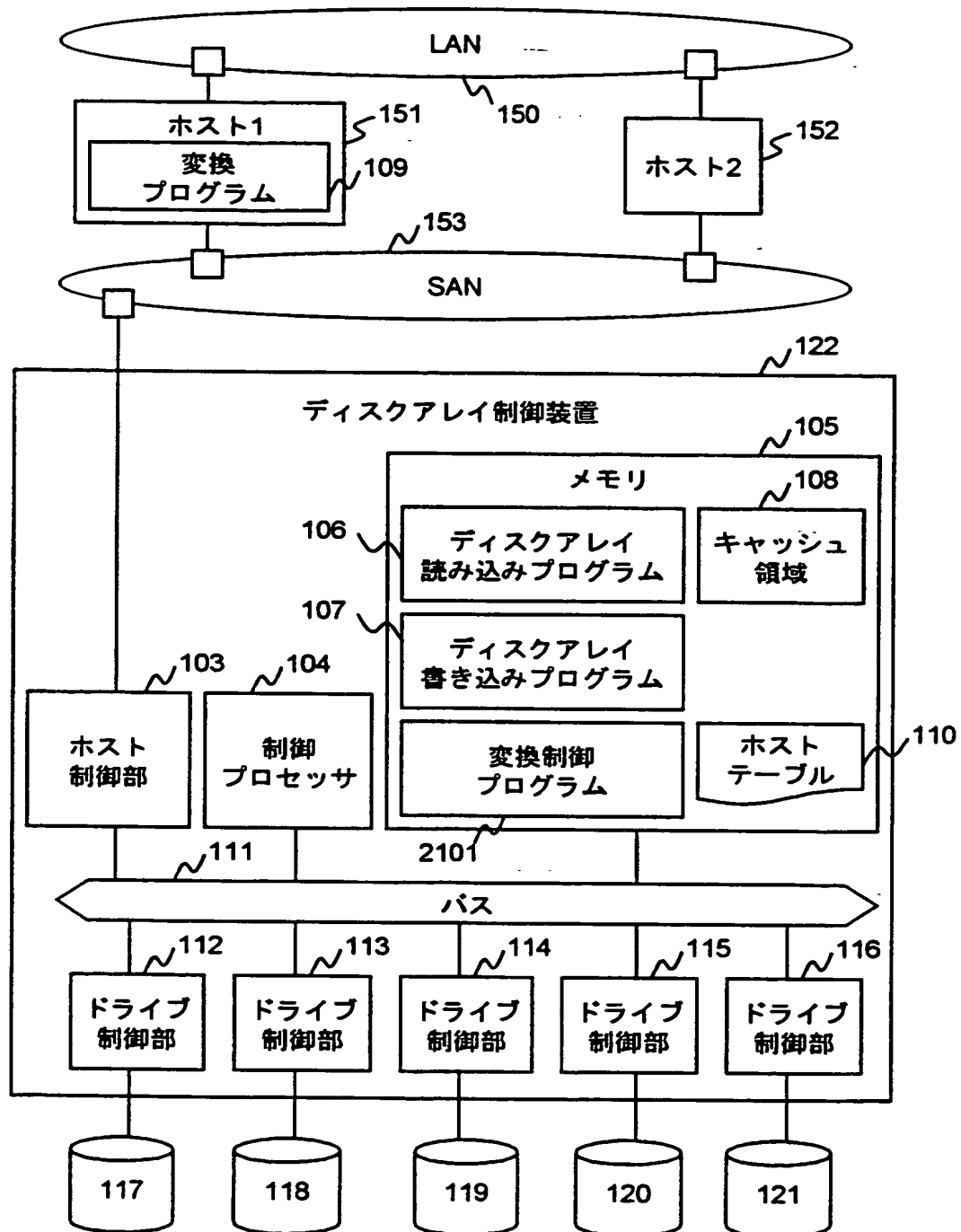
## 第 2 0 図



**THIS PAGE BLANK (USPTO)**

21 / 39

第 2 1 図

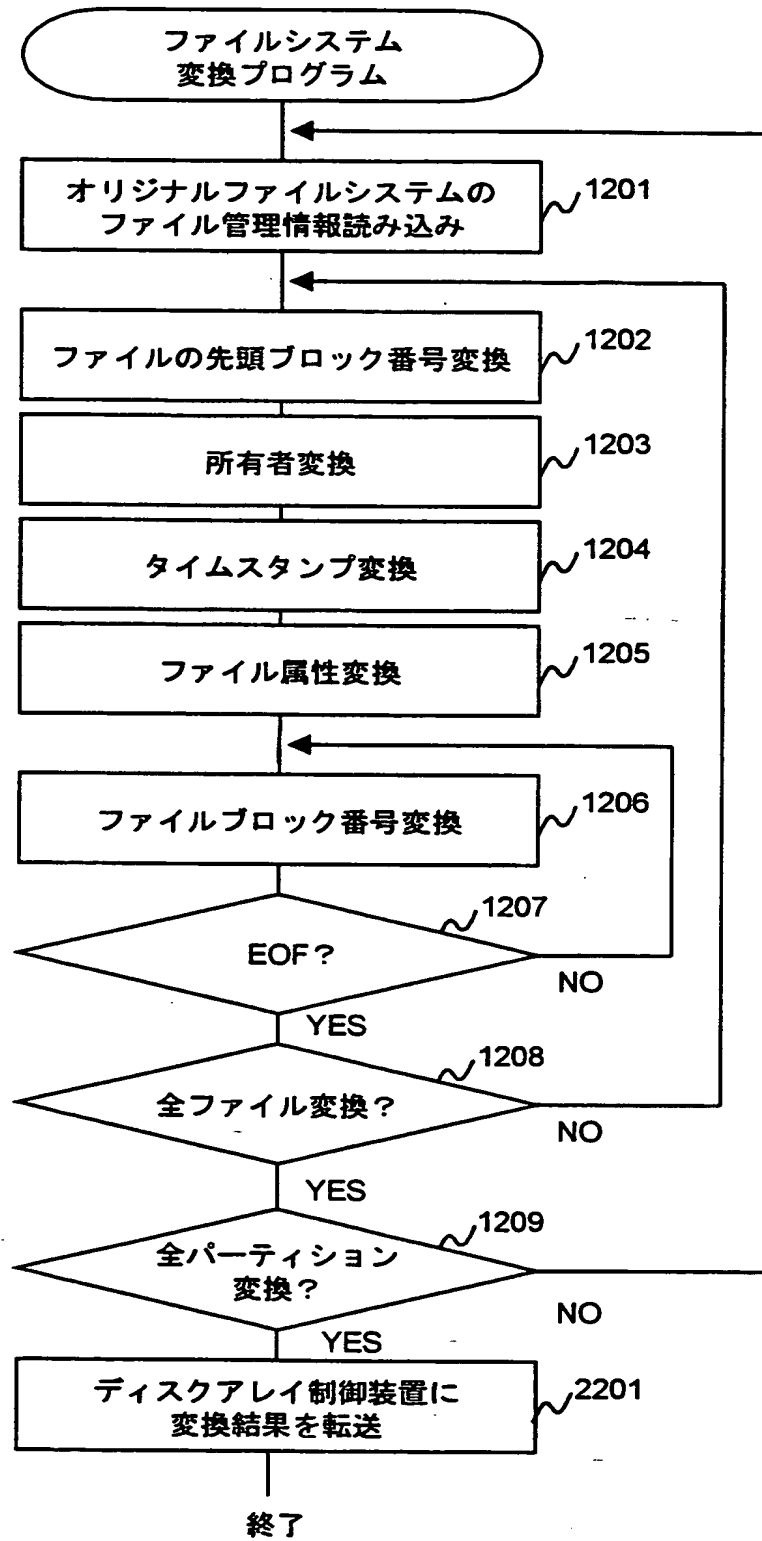


**THIS PAGE BLANK (USFC,**



22 / 39

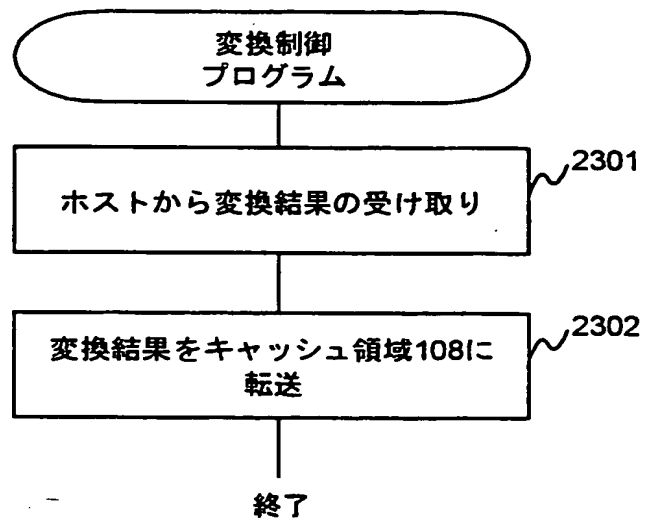
## 第 2 2 図



**THIS PAGE BLANK (USPTO)**

23 / 39

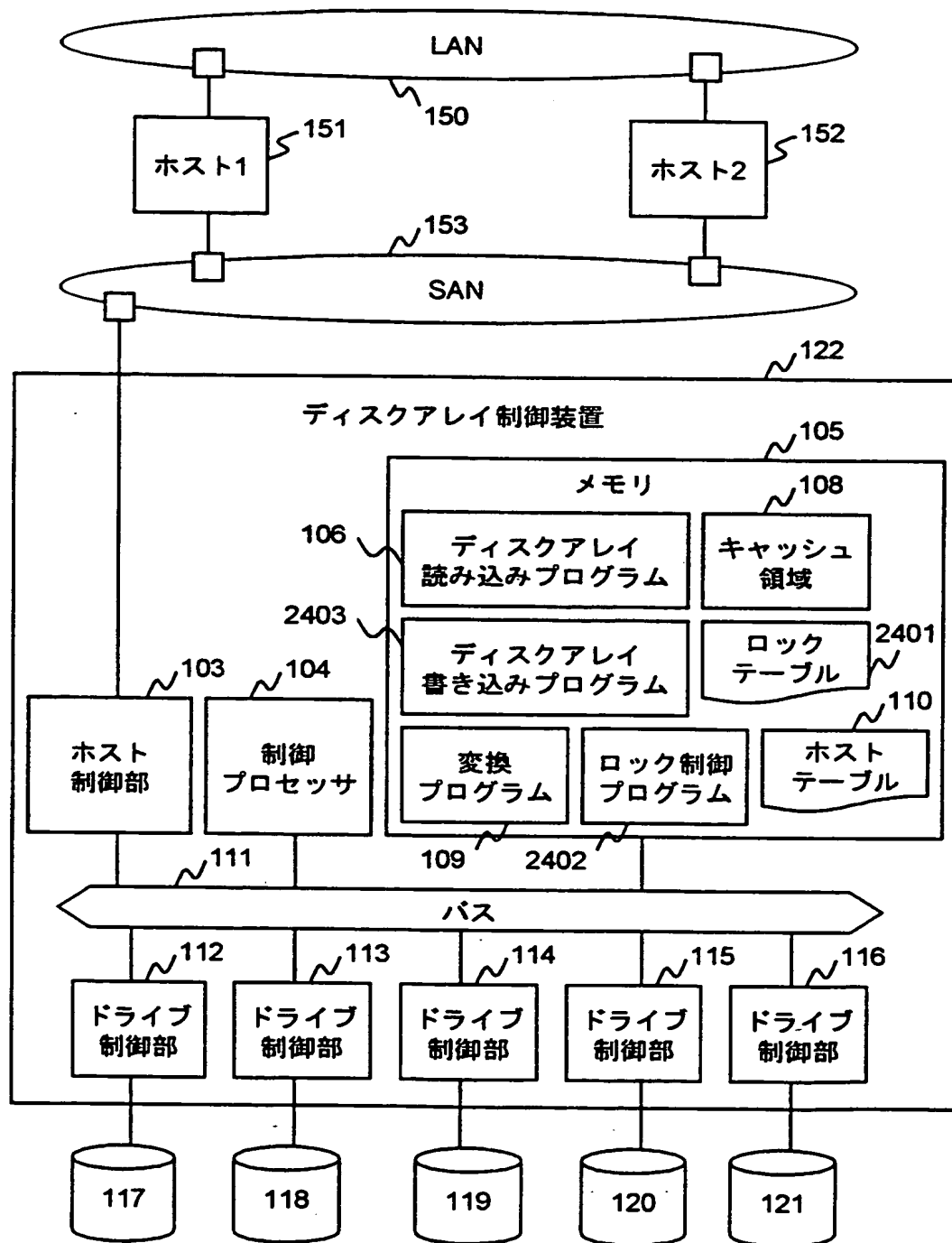
## 第 2 3 図



**THIS PAGE BLANK (USPTO)**

24 / 39

第 2 4 図



**THIS PAGE BLANK (USP16)**

25 / 39

## 第 2 5 図

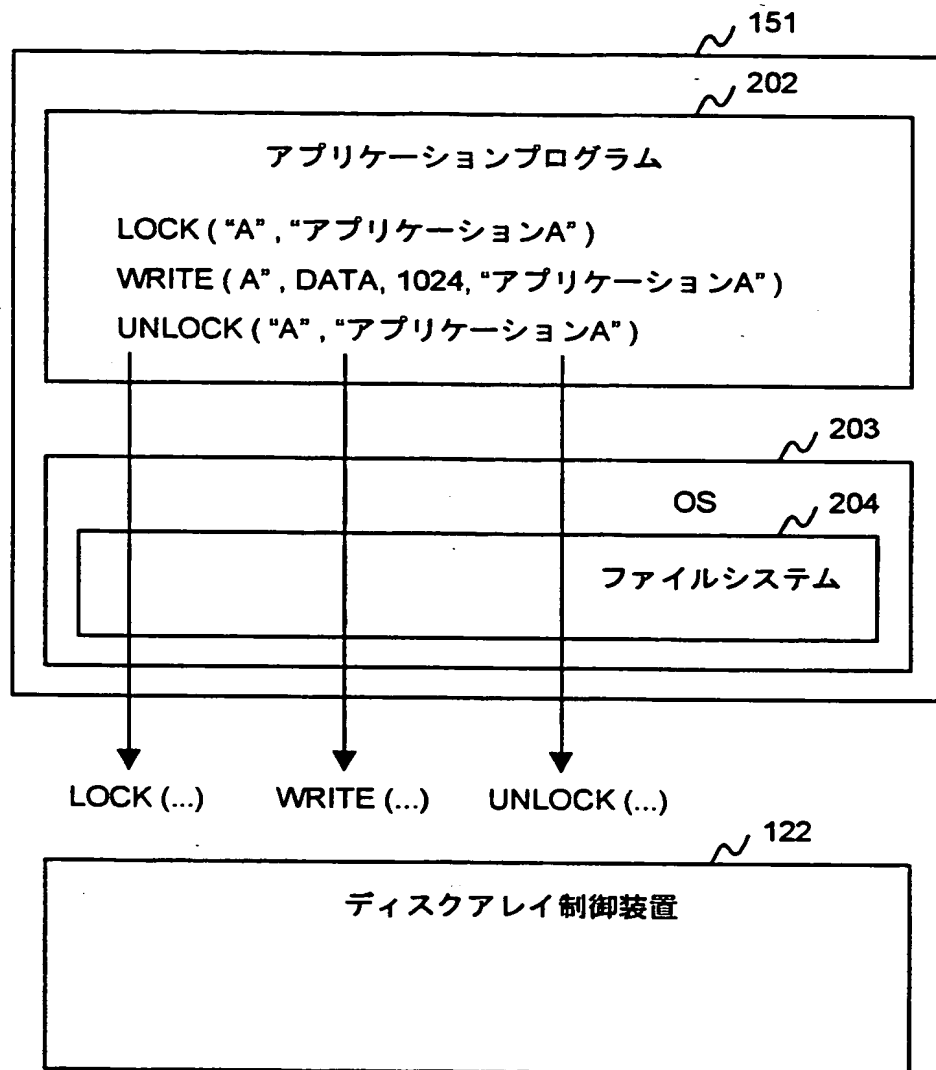
2501	2502	2503	2504	2505
ファイル名	LOCK/ UNLOCK	ホストID	ロック保持者	ロック回数
A	LOCK	ホストA	アプリケーションA	1
B	UNLOCK			0
C	LOCK	ホストA	アプリケーションB	2
D	UNLOCK			0
⋮	⋮	⋮		⋮

**THIS PAGE BLANK (USPTO)**



26 / 39

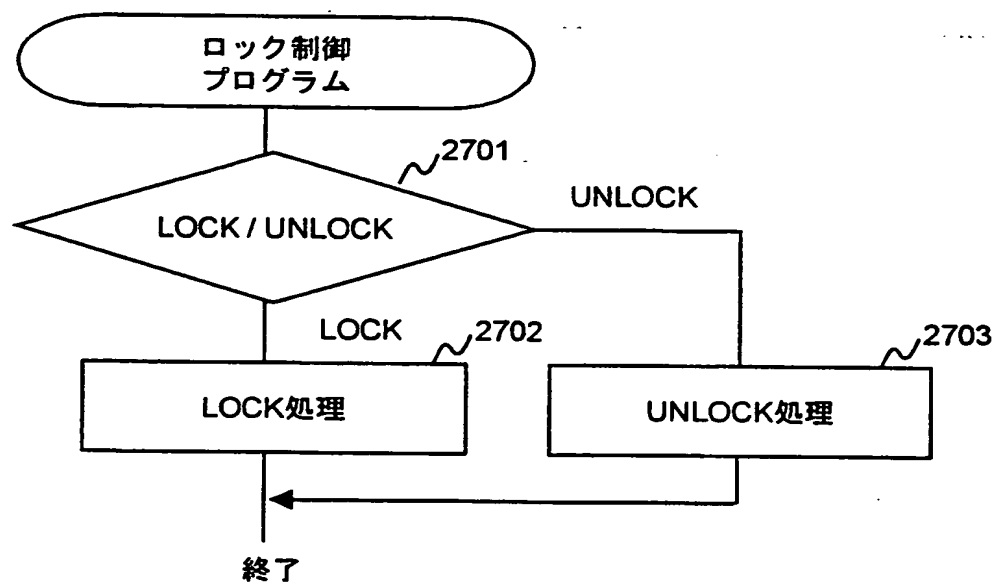
第 2 6 図



**THIS PAGE BLANK (USPTO)**

27 / 39

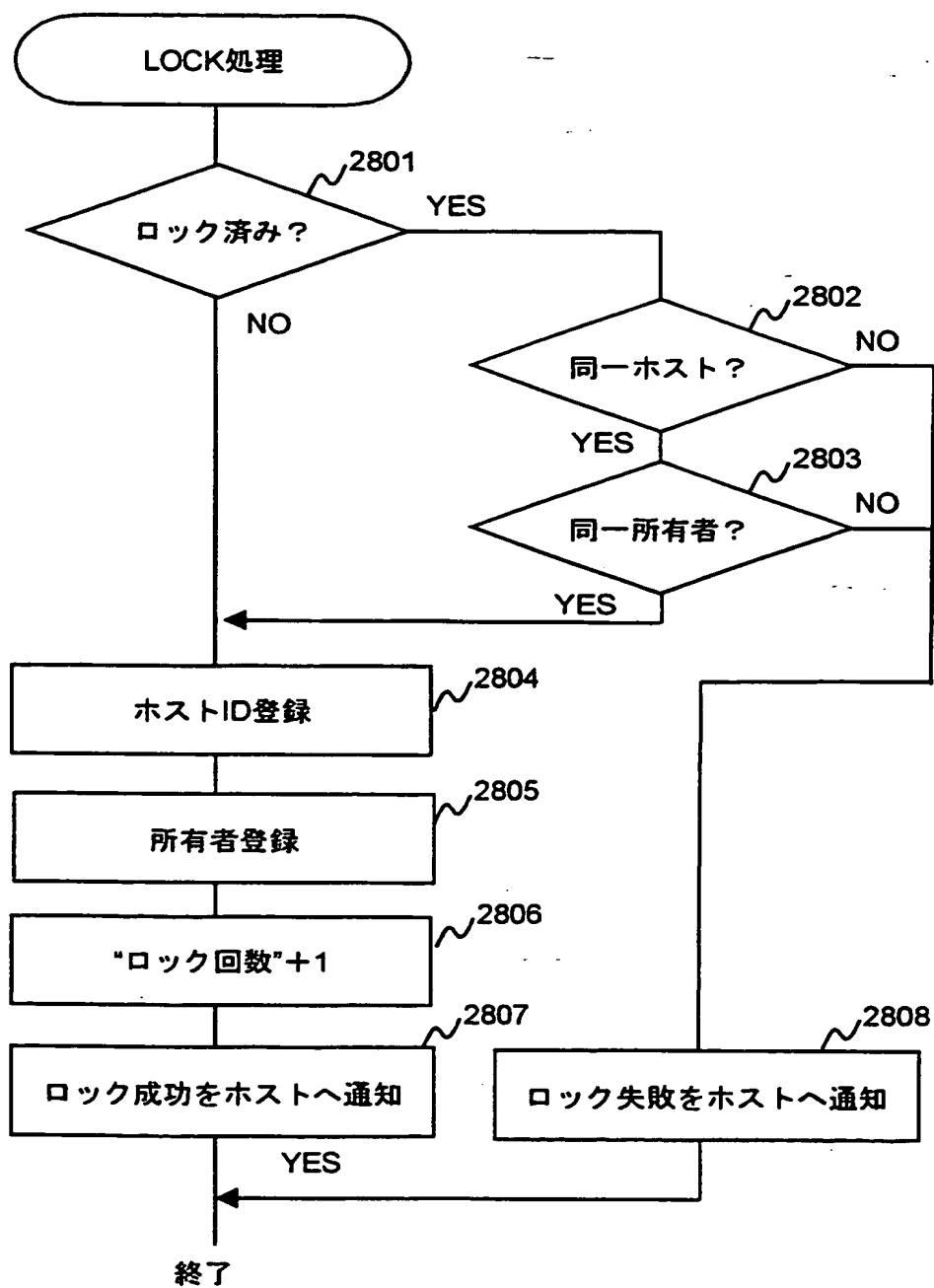
## 第 27 図



**THIS PAGE BLANK (USPTO)**

28 / 39

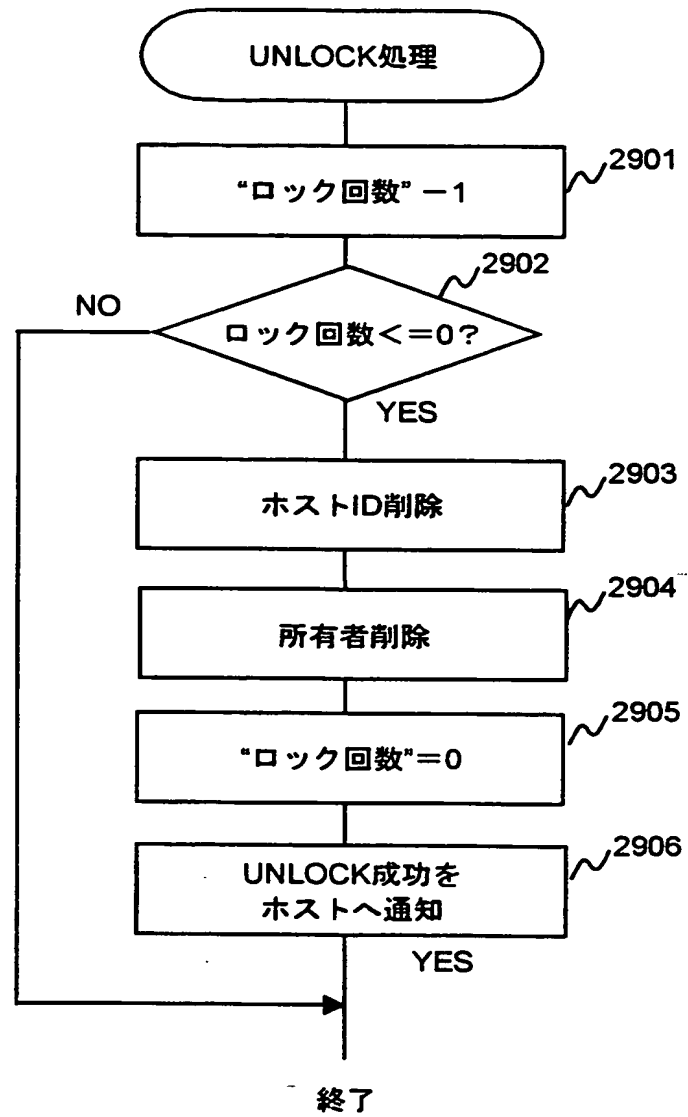
第28図



**THIS PAGE BLANK (USPTO)**

29 / 39

## 第 2 9 図

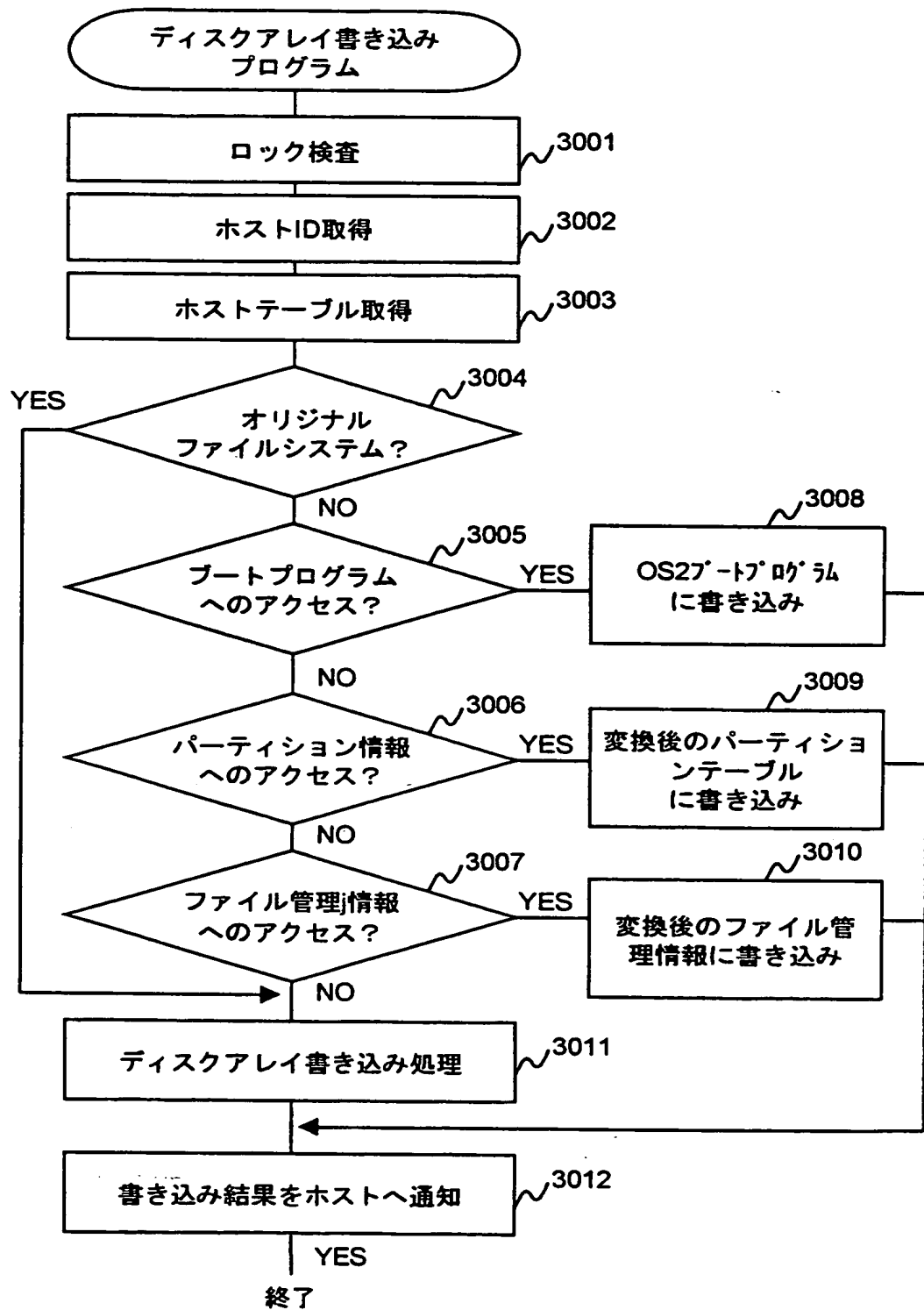


**THIS PAGE BLANK (USPTO)**



30 / 39

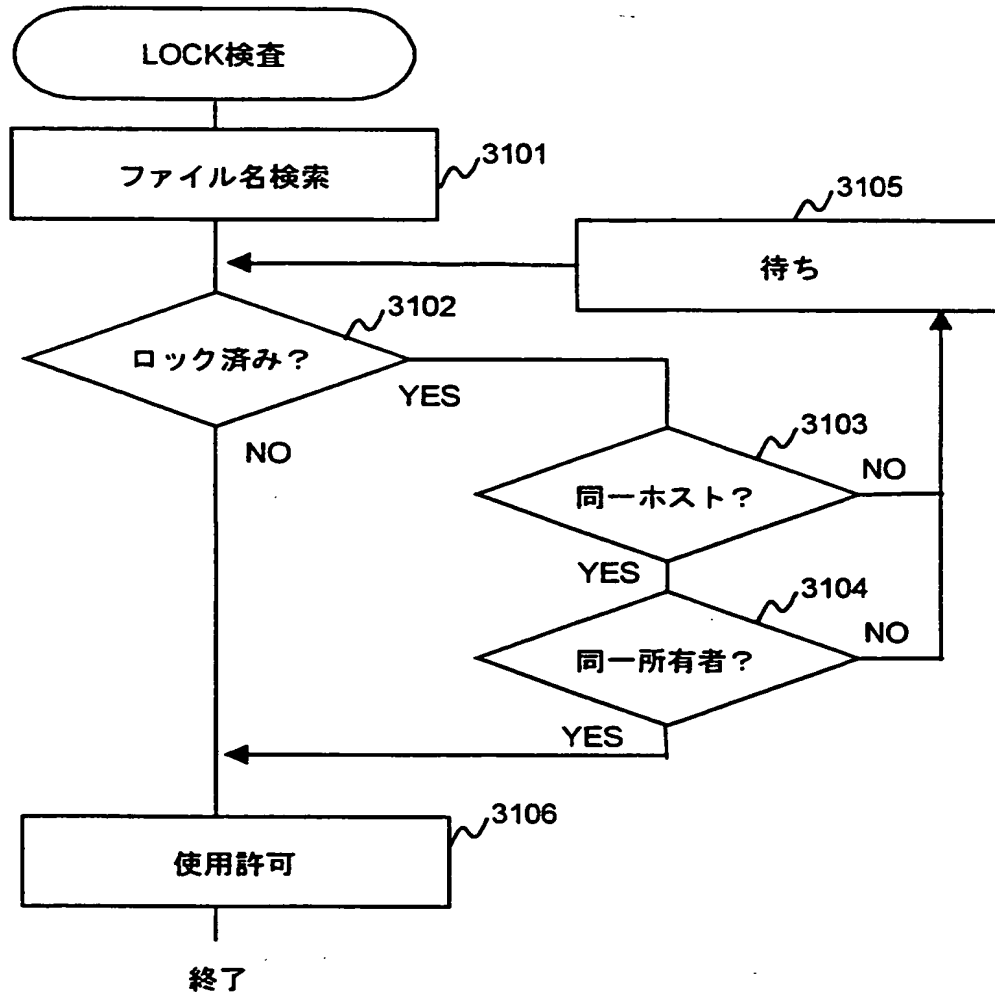
## 第30図



**THIS PAGE BLANK (USPTO)**

31 / 39

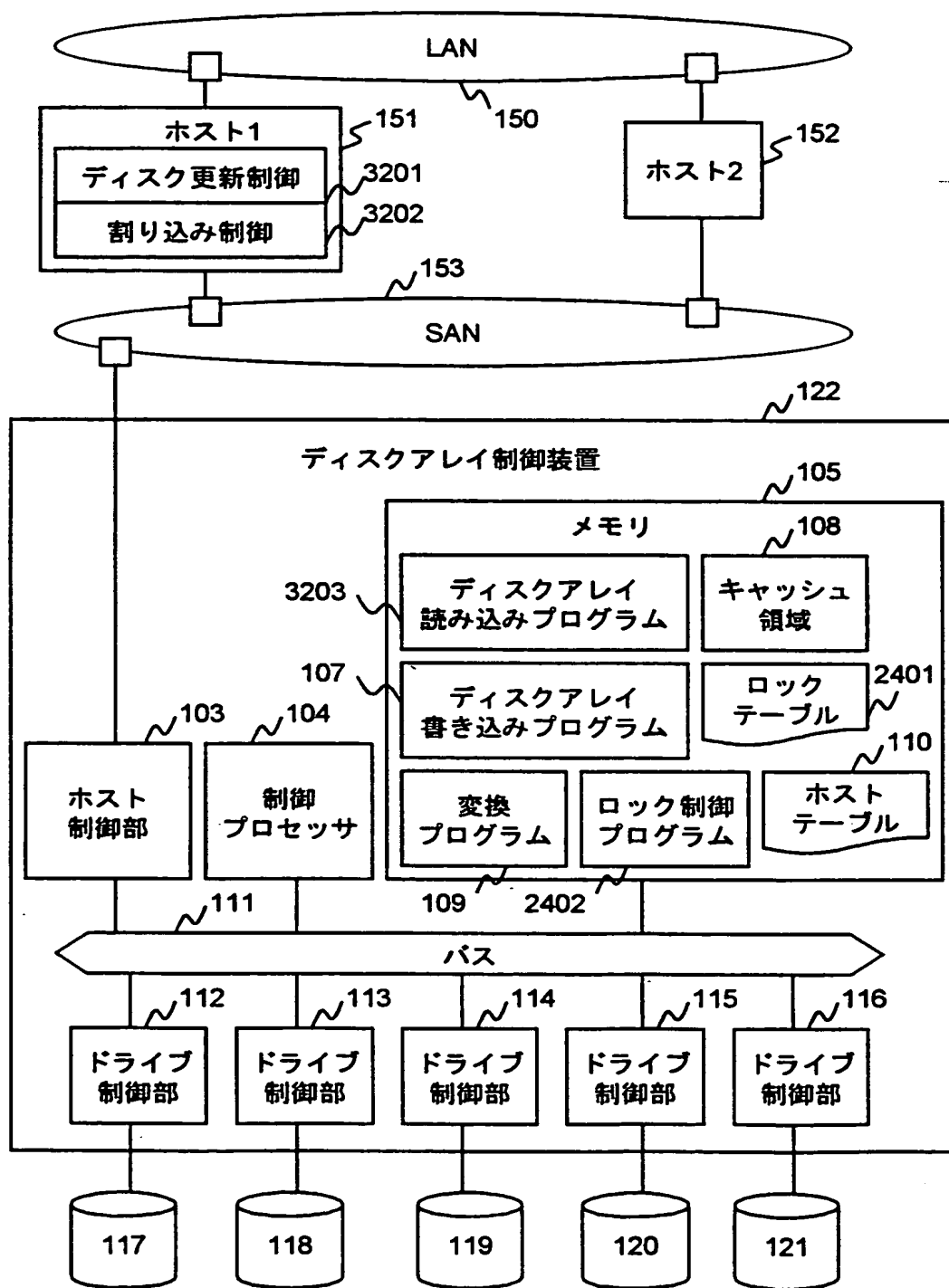
## 第 3 1 図



**THIS PAGE BLANK (USPTO)**

32 / 39

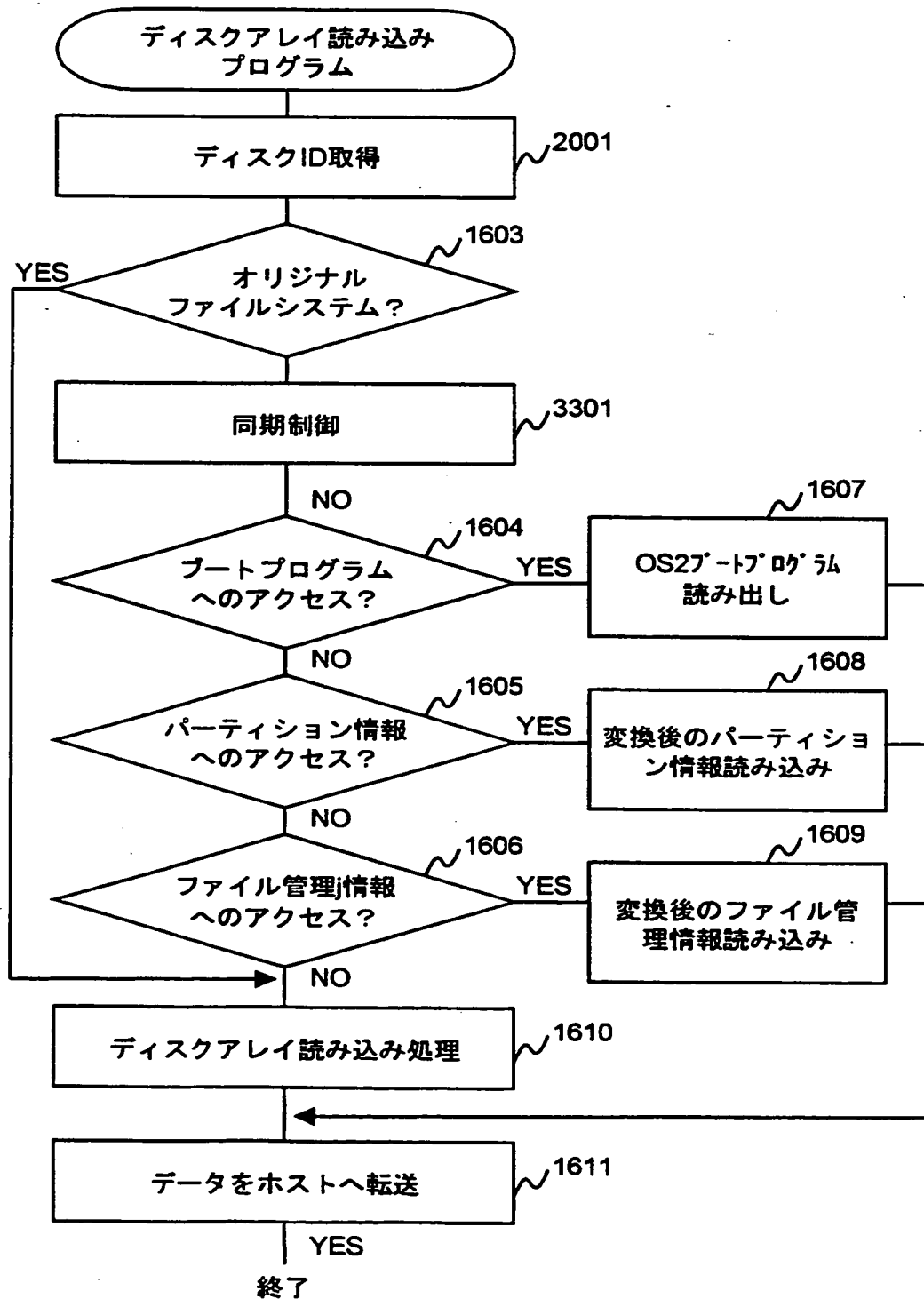
第 3 2 図



**THIS PAGE BLANK (USPTO)**

33 / 39

## 第 3 3 図

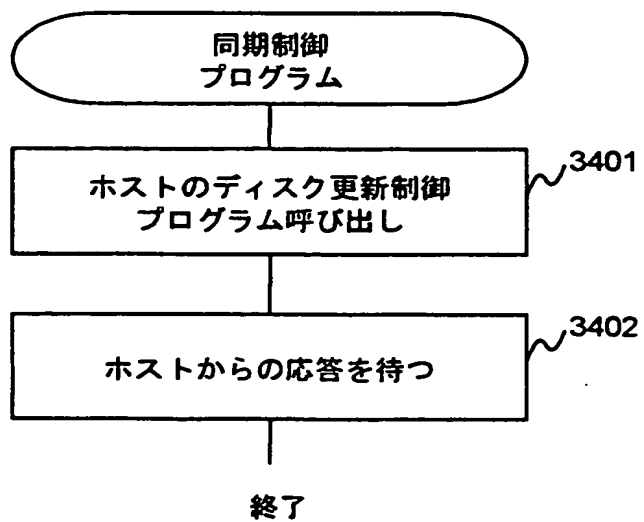


**THIS PAGE BLANK (USPTO)**



34 / 39

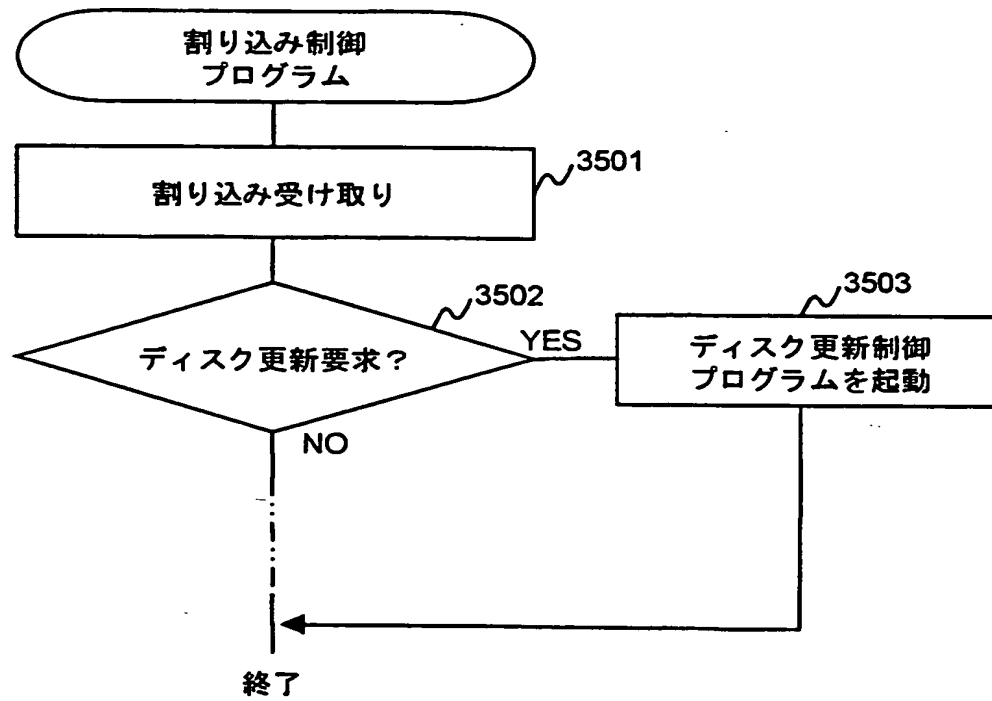
## 第 3 4 図



**THIS PAGE BLANK (USPTO)**

35 / 39

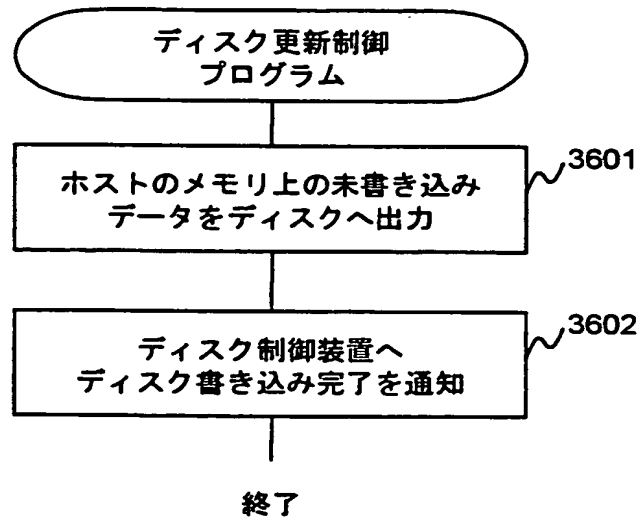
## 第 3 5 図



**THIS PAGE BLANK (USPTO)**

36 / 39

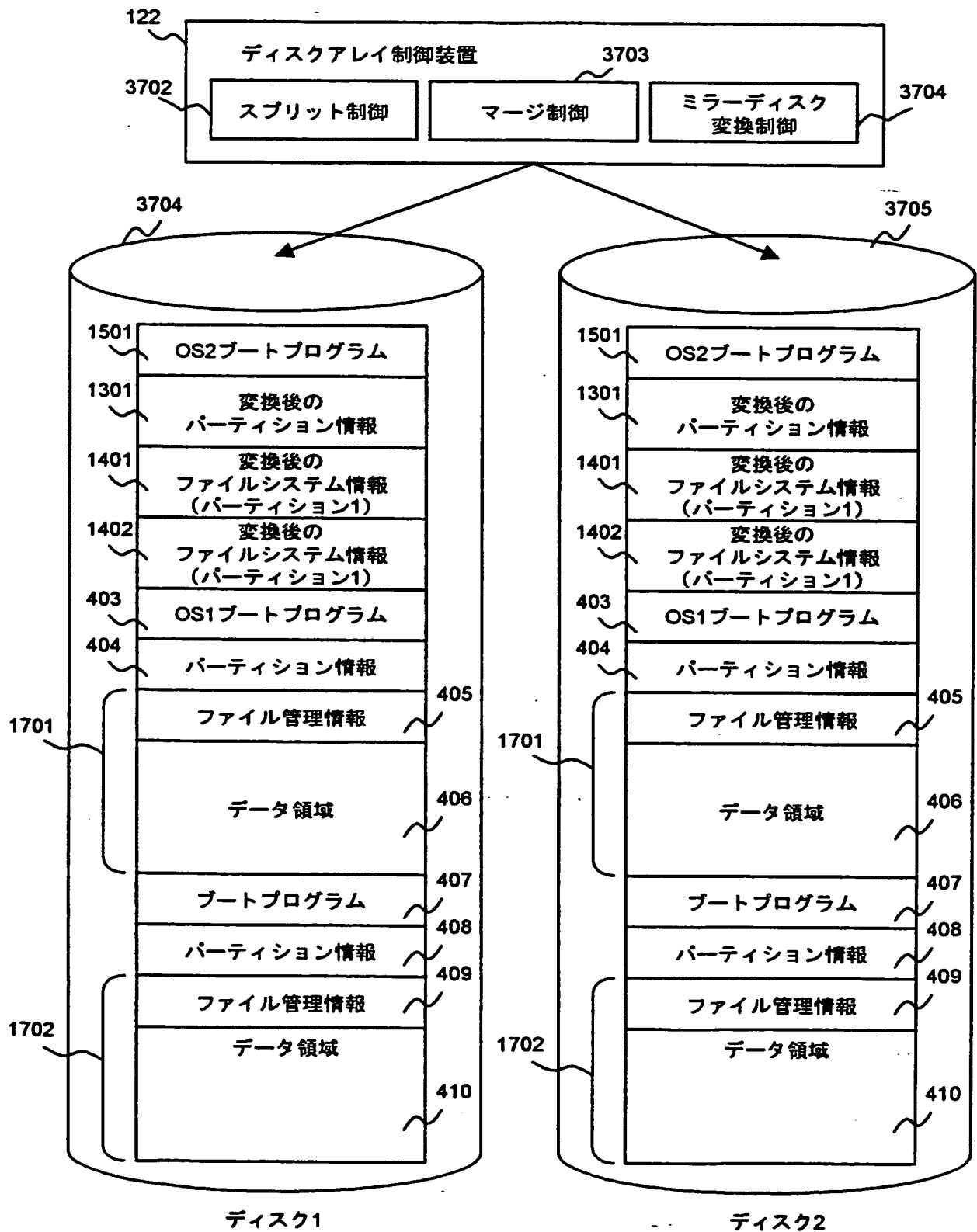
第 3 6 図



**THIS PAGE BLANK (USPTO)**

37 / 39

第 3 7 図

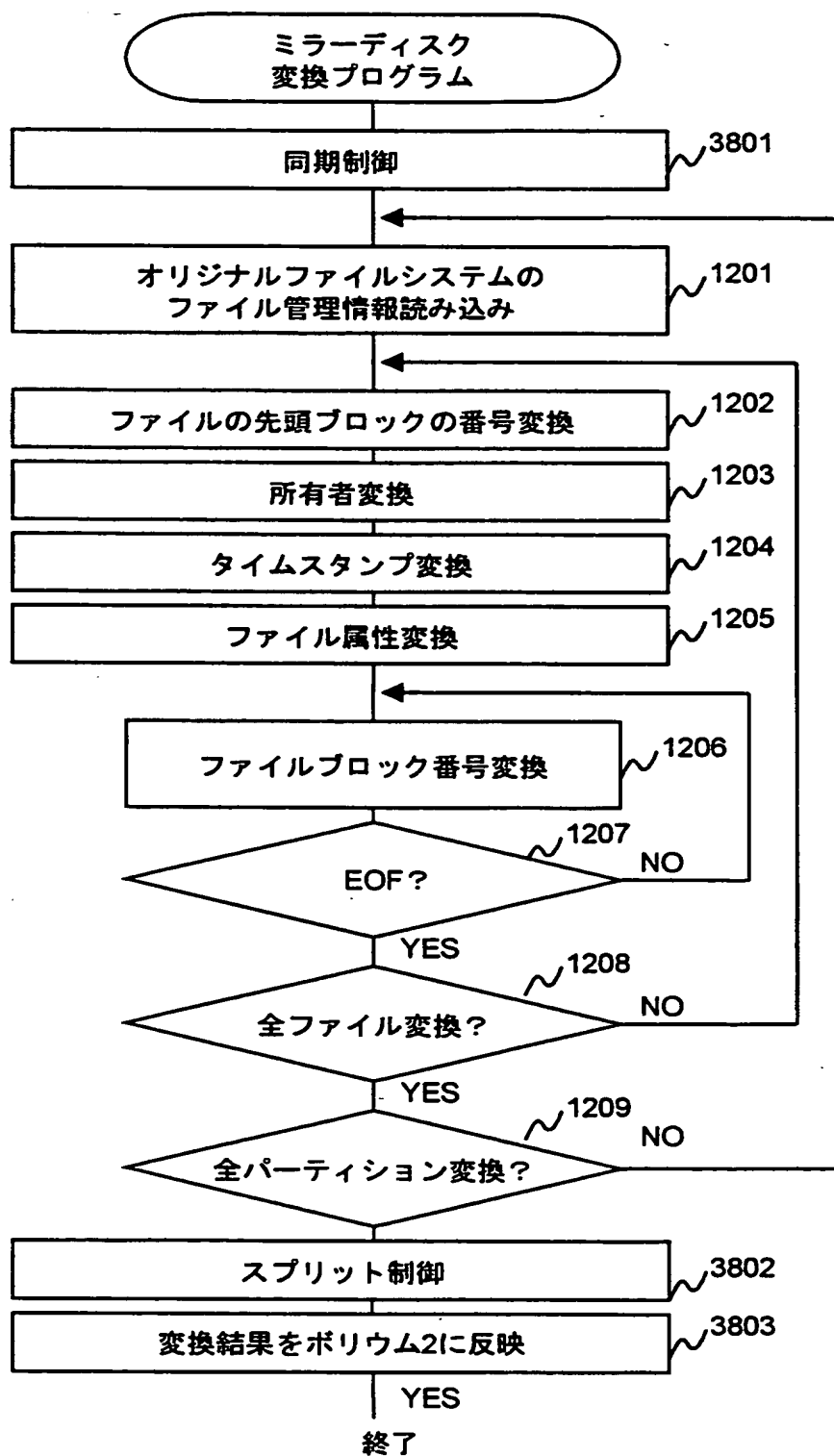


**THIS PAGE BLANK (USPTO)**



38 / 39

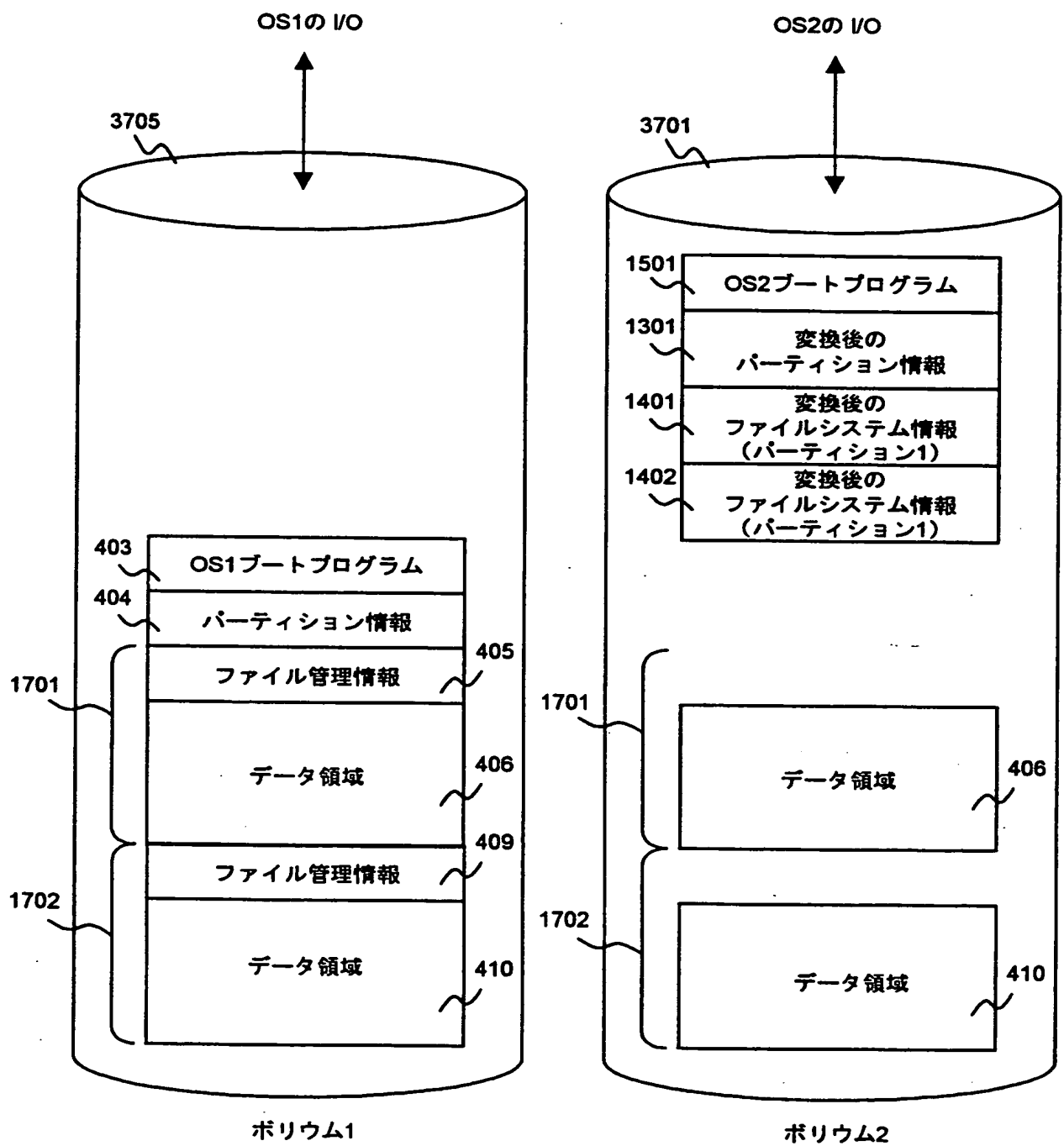
第 3 8 図



**THIS PAGE BLANK (USPTO)**

39 / 39

第 3 9 図



**THIS PAGE BLANK (USPTO)**

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP99/05357

A. CLASSIFICATION OF SUBJECT MATTER  
Int.Cl<sup>7</sup> G06F12/00

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl<sup>7</sup> G06F12/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	Computopia, 33[380] (1998-05-01), Kabushiki Kaisha Computer Age K.K. "Storage Sensen ni Ijou ari; Java ga Storage wo kaeru Multi Platform Storage ni Honkaku Sannyuu shita SAN no Senryaku", pp.34-37	1-5 6-9
X Y	JP, 10-254742, A (Hitachi, Ltd.), 25 September, 1998 (25.09.98) (Family: none)	1,2 3-9
X Y	JP, 9-325905, A (Hitachi, Ltd.), 16 December, 1997 (16.12.97) & EP, 811905, A & US, 5920893, A	1,2,4,5 3,6-9
Y	JP, 5-265661, A (Hitachi, Ltd.), 15 October, 1993 (15.10.93) (Family: none)	1-9
Y	JP, 9-128277, A (Chubu Nippon Denki Software K.K.), 16 May, 2016 (16.05.16) (Family: none)	1-9
Y	JP, 9-179758, A (Fujitsu Limited), 11 July, 1997 (11.07.97) (Family: none)	6,7

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

## \* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&amp;" document member of the same patent family

Date of the actual completion of the international search  
28 December, 1999 (28.12.99)Date of mailing of the international search report  
25 January, 2000 (25.01.00)Name and mailing address of the ISA/  
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

**THIS PAGE BLANK (USPTO)**

## 国際調査報告

国際出願番号 PCT/J P 99/05357

## A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl<sup>7</sup>G 06 F 12/00

## B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl<sup>7</sup>G 06 F 12/00

最小限資料以外の資料で調査を行った分野に含まれるもの

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

## C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X Y	Computopia、33[380](1998-05-01)株式会社コンピュータ・エージ社「ストレージ戦線に異状あり Javaがストレージを変える マルチプラットフォーム・ストレージに本格参入したサンの戦略」p. 34-37	1-5 6-9
X Y	J P, 10-254742, A (株式会社日立製作所) 25.9月.1998(25.09.98), (ファミリーなし)	1, 2 3-9
X Y	J P, 9-325905, A (株式会社日立製作所) 16.12月.1997(16.12.97)& E P, 811905, A&US, 5920893, A	1, 2, 4, 5 3, 6-9
Y	J P, 5-265661, A (株式会社日立製作所) 15.10月.1993(15.10.93), (ファミリーなし)	1-9
Y	J P, 9-128277, A (日本電気ソフトウェア株式会社) 16.5月.1997(16.05.97), (ファミリーなし)	1-9

☒ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

## \* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの  
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの  
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)  
「O」口頭による開示、使用、展示等に言及する文献  
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&amp;」同一パテントファミリー文献

国際調査を完了した日

28.12.99

国際調査報告の発送日

25.01.00

国際調査機関の名称及びあて先

日本国特許庁 (ISA/J P)

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

小田 浩



5N

9188

電話番号 03-3581-1101 内線 6904

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
Y	J P, 9-179758, A (富士通株式会社) 11.7月.1997(11.07.97), (ファミリーなし)	6, 7